第1章　MATLAB基本介绍

本章介绍了MATLAB的产生背景，列举了MATLAB金融工具箱基本内容。国外很多著名的金融机构利用MATLAB进行金融产品的设计和风险管理，MATLAB已经成为金融工程解决方案之一。

1.1　MATLAB数值计算特点

1.1.1　MATLAB产生背景

MATLAB诞生于20世纪70年代，是由美国新墨西哥大学前计算机系主任Cleve Moler博士创立的。MATLAB是矩阵(Matrix)和实验室(Laboratory)两个英文单词前3个字母拼写而成的。1983年春天，Cleve Moler在斯坦福大学访问时，结识了软件工程师Jack Little，随后Cleve Moler、Jack Little、Steve Bangert用C语言共同开发了第二版MATLAB，使得MATLAB不仅具有数值计算功能，而且具有可视化功能。

1984年，Cleve Moler和Jack Little发起成立了MathWorks公司，专门研究商业版MATLAB。1993年MathWorks公司推出MATLAB 4.0版本，1995年又推出了MATLAB 4.2C版本，1997年推出MATLAB 5.0，2000年10月推出MATLAB 6.0，2002年8月MathWorks公司向市场推出了MATLAB 6.5。MATLAB每一次版本升级都增加了大量数值计算功能，使其数值计算、符号计算和图形处理功能大大加强，界面越来越友好；2005年8月推出了MATLAB 7.10。2007年5月18日MathWorks于北京成立了中国分公司，也是其全球第十二家国外分公司，致力于拓展中国市场。

自20世纪90年代以来，MATLAB成为国际公认的优秀计算软件，在大学及业界得到广泛运用。在欧美大学中，应用代数、数理统计、自动控制、电子信号处理、模拟和数字通信、时间序列分析及动态系统仿真等课程都把MATLAB作为配套教材，成为本科生、硕士生、博士生必备的基本工具。

MATLAB长于数值计算，能处理大量数据，而且运算效率很高，MathWorks公司在此基础上拓展了符号计算、文字处理、可视化建模和实时控制能力。MATLAB作为计算工具和科技资源，可以扩大科学研究范围，提高工程技术效率。

1.1.2 MATLAB 语言的优点

1. 功能强大

MATLAB 4.0 以上各个版本，不仅在数值计算上保持领先，而且开发出了自己的符号运算系统，功能上不逊于 MatCAD，Mathematic 等软件，让研究人员摆脱了烦琐的软件学习过程，可以进行矩阵变换、多项式运算、微积分运算、线性与非线性方程求解、常微分方程求解、插值与数值拟合、统计、时间序列、金融衍生产品定价、资产组合分析、固定收益证券定价和风险管理等。MATLAB 产品分为以下几大类：

- 数据分析；
- 数值和符号计算；
- 工程与科学绘图；
- 控制系统设计；
- 数字图像信号处理；
- 金融计算；
- 建模、仿真及模型开发；
- 应用开发；
- 图形、图像界面开发。

2. 简单易学

一般语言编写程序、调试程序需要经过五个步骤：编辑、编译、连接、执行、调试，各个步骤之间是顺序关系，编程过程就是在它们之间进行循环。MATLAB 语言与其他语言相比，较好地解决了上述问题。MATLAB 软件是解释性语言，调试程序的手段非常丰富，速度也快，把编辑、编译、连接和执行融为一体，在同一界面上操作，可以快速查出输入程序中的书写错误、语法错误，提高了用户编写程序的效率。MATLAB 在运行时，可以直接输入命令语句，包括调用 M 文件语句，每输入一条命令，程序就会立即对其进行处理，完成编译、连接、执行全过程。再如，将 MATLAB 源程序编译为 M 文件，编辑后源程序可以立即执行，如果有错，则用户界面上也会给出详细的出错信息。MATLAB 软件易懂易学，允许数学形式编写代码，比 BASIC、FORTRAN 和 C 语言更加符合书写习惯，它的操作符和功能函数就是数学上的简单的英文表达式。MATLAB 还拥有强大的帮助系统，可以查询到各种命令的使用说明和详细案例；为便于初学者快速了解 MATLAB 功能，还提供了演示窗口。

3. 高效便捷的矩阵和数组运算

MATLAB 语言与 BASIC、FORTRAN 和 C 语言一样都规定了矩阵算术运算、关系运算

符、逻辑运算符、条件运算符和赋值运算符,而且这些运算符适用于数组运算。对变量定义维数,并给出矩阵函数、特殊矩阵专用库函数,使之在处理信号分析、数学建模、系统识别、控制优化等领域的问题时变得非常简单,MATLAB有望成为名副其实的"万能演算稿纸"似的科学计算语言。

4. 可扩充能力强,适于二次开发

MATLAB的快速发展得益于非常好的营运模式。MATLAB是一个开放系统,具有非常好的可扩充性和可开发性,用户可以非常方便地看到源程序,可以对源程序进行修改,创建符合自己需求的文件库。对外而言,MATLAB并不具有排他性,可以和FORTRAN、C语言、Visual Basic通过接口相连接,可以方便地相互调用程序。MATLAB语言具有丰富的库函数,进行复杂的数学运算时可以直接调用,因此用户可以根据自身需要建立或扩充新的库函数。良好的交互性使得程序员可以利用可重复使用程序代码,提高效率。

5. 移植性好

MATLAB是用C语言编写的,它继承了C语言可移植性好的特点,因此可以移植到C语言操作平台上。MATLAB适用的操作系统有Windows、UNIX和Linux等,除了内部函数外,MATLAB的核心文件和工具箱都是公开的,读者可以对源文件进行修改,使其更加符合自己的需要。

6. 方便的绘图功能

MATLAB的绘图功能十分强大,它包含一系列绘图命令,例如线性坐标、对数坐标、半对数坐标及极坐标,只要调用不同的绘图函数即可。图中的标题、坐标轴标注以及网格绘制仅需调用相应的命令,非常简单,并可以绘制不同颜色的点、线、复线和多重线。

当前,MATLAB软件因为其良好的开放性和稳定性,在工程计算中得到广泛应用。最近10年来,MATLAB已经成为国际公认的标准计算软件,全球超过500 000个工程师与科学家使用MATLAB软件,2 000多家金融机构使用MATLAB软件建立经济、金融模型,帮助他们估计经济中的各种风险,并对风险进行有效管理,提高企业经营效率。

1.2 系统的金融工程解决方案

MATALB的金融工具箱是一个函数库,每个函数就是一个金融问题的解决方案,如果把这些函数组合起来就变成一个程序,把程序集结起来就变成一个模块和系统。

1.2.1 MATLAB金融工具箱模块

1. Financial Toolbox

MATLAB自带金融工具箱，具有下列功能：

- 日期数据处理；
- 资产均值-方差分析；
- 时间序列分析；
- 固定收益计算；
- 有价证券的收益和价格；
- 统计分析；
- 定价和灵敏度分析；
- 年金和现金流计算；
- 抵押支持债券。

2. Financial Derivatives Toolbox

Financial Derivatives Toolbox是金融衍生产品工具箱，用于固定收益、金融衍生物以及风险投资评估分析，也可用于各种金融衍生物定价策略以及敏感度分析。

3. Financial Time Series Toolbox

Financial Time Series Toolbox用于分析金融市场的时间序列数据。金融数据是时间序列数据，例如股票价格或每天的利息波动，可以用该工具箱进行更加直观的数据管理。该工具箱支持下列功能：① 提供两种创建金融时间序列的对象（用构造器和转换文本文件）；② 可视化金融时间序列对象；③ 技术分析函数分析投资。

4. Fixed - Income Toolbox

Fixed - Income Toolbox扩展了MATLAB在金融财经方面的应用，可以用固定收益模型进行计算，例如定价、收益和现金流动等有价证券的固定收益计算。支持的固定收益类型包括有价证券抵押回报、社会债券和保证金等。该工具箱还能够处理相应金融衍生物的计算，支持抵押回收有价证券、国债和可转换债券等的计算。

5. GARCH Toolbox

GARCH Toolbox提供了一个集成计算环境，允许对单变量金融时序数据的易变性进行建模。GARCH Toolbox使用一个广义ARMAX/GARCH复合模型对带有条件异方差的金

融时序数据进行仿真、预测和参数识别。GARCH Toolbox提供了基本工具为单变量广义自回归条件异方差GARCH(Generalized Auto Regressive Conditional Heteroskedasticity)易变性进行建模。GARCH Toolbox采用单变量GARCH模型对金融市场中的变化性进行分析。

上述工具箱基本上囊括了通常的金融计算，适用于金融学术研究，特别适合金融实务工作者进行金融计算。Financial Toolbox提供了一个基于MATLAB的财务分析支撑环境，可以完成许多种财务分析统计任务；从简单计算到全面的分布式应用，财务工具箱都能够用来进行证券定价、资产组合收益分析、偏差分析和优化业务量等工作。

1.2.2 使用MATLAB的主要金融机构

使用MATLAB的主要金融机构如下：

美国三大评级机构；

美国前十五大资产管理公司[①]；

美国前十大商业银行中有9家使用MATLAB[②]；

美国前十五大对冲基金中有12家使用MATLAB[③]；

OECD国家的85 %其中央银行使用MATLAB。

1.2.3 MATLAB网上资源

下面是部分网站MATLAB金融资源：

1) Mathworks公司主页

http://www.mathworks.com/

2) MathWorks中国分公司主页

http://www.mathworks.cn/index.html? ref=pt

3) MATLAB程序交流中心

http://www.mathworks.com/matlabcentral/

4) MATLAB金融程序交流中心

http://www.mathworks.com/matlabcentral/link_exchange/MATLAB/Finance_and_Economics/index.html

5) MATLAB定量金融分析程序代码

http://www.mathfinance.cn/showcategory.php? cid=20

6) MATLAB程序交流网站

http://sourceforge.net/projects/mlmechtrade/

---

① 根据Institutional Investor杂志2006年度排名。

② 根据American Bank的排名。

③ 根据Alpha Magazine杂志2006年度排名。

7）MATLAB程序交流网站

http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do?objectType=category&objectId=3

8）MATLAB编写的计量经济学教程

http://www.spatial-econometrics.com/

9）MathWorks公司网上授课

http://www.mathworks.cn/company/events/archived_webinars.html?sec=finance

10）威廉·夏普的宏观投资分析网站

http://www.stanford.edu/~wfsharpe/mia/mia.htm

11）金融衍生品程序网站

http://www.global-derivatives.com/index.php

1.2.4 MATLAB安装组件

安装MATLAB时将安装盘放入光驱，系统会自动搜索到Autorun文件并进入安装界面。当用户填写完注册信息和使用协议后，进入选择安装界面。表1.1是MATLAB安装时可以选择的组件。

表1.1 MATLAB可供安装的组件

组件名称	功　能
MATLAB	MATLAB主程序
Simulink	动态仿真
CDMA Reference Blockset	CDMA参考模块
Communications Blockset	通信模块
Communications Toolbox	通信工具箱
Control SystemToolbox	控制系统工具箱
DSP Blockset	数据信号模块
Data Acquisition Toolbox	数据采集工具箱
Database Toolbox	数据库工具箱
Datafeed Toolbox	数据载入工具箱
Dials and Gauges Blockset	刻度标尺模块
Excelink	Excel链接
Filter Design Toolbox	滤波设计工具箱
Financial Derivatives Toolbox	金融衍生品工具箱
Financial TimeSeries Toolbox	金融时间序列工具箱

续表 1.1

组件名称	功 能
Financial Toolbox	金融工具箱
Fixed - Point Blockset	固定点模块
Fuzzy Logic Toolbox	模糊逻辑工具箱
GARCH Toolbox	GARCH 工具箱
Image Processing Toolbox	图像处理工具箱
Instrument Control Toolbox	器材控制工具箱
LMI Control Toolbox	LMI 控制工具箱
MATLAB C/C++ Garphics Library	MATLAB C/C++图形库
MATLAB C/C++ Math Library	MATLAB C/C++数学库
MATLAB Compiler	MATLAB 编译器
MATLAB report generator	MATLAB 报告生成器
Mapping Toolbox	制图工具箱
Model Predictive Control Toolbox	模型预测控制工具箱
Motorola DSP Developer Kit	摩托罗拉数字信号开发工具箱
Mu - Analysis and Symthesis Toolbox	Mu 分析与合成工具箱
Neural Network Toolbox	神经网络工具箱
Nonlinear Control Design Blockset	非线性设计控制系统
Optimization Toolbox	优化工具箱
Partical Differential Equation Toolbox	偏微分方程工具箱
Power System Blockset	动力系统模块
Real - Time Workshop	实时工作间
Real - Time Workshop Ada Coder	实时工作间内置编码
Requirements Management Interface	需求管理界面
Robust Control Toolbox	鲁棒控制工具箱
SB2SL (Converts Models to Simulink)	模型转换成 Simulink 工具
Signal Processing Toolbox	信号处理工具箱
Simulink report Generator	Simulink 报表生成器
Spline Toolbox	样条工具箱
Stateflow	状态流工具箱
Stateflow Code	状态流控制代码
Statistis Toolbox	统计工具箱

续表 1.1

组件名称	功　能
Symbolic Math Toolbox	符号运算工具箱
System Identification Toolbox	系统识别工具箱
Wavelet Toolbox	小波工具箱
xPC Target	xPC 对象
xPC Target Embedded Option	xPC 对象内置属性

金融初学者可以选择如下组件：

MATLAB；

Database Toolbox；

Datafeed Toolbox；

Excelink；

Financial Derivatives Toolbox；

Financial TimeSeries Toolbox；

Financial Toolbox；

GARCH Toolbox；

Spline Toolbox；

Optimization Toolbox；

Statistis Toolbox；

Symbolic Math Toolbox。

读者也可以根据需要添加其他相关工具箱，安装完成后重新启动计算机，然后直接运行MATLAB。

思考题

1. 查找 MATLAB 相关网站，学习如何使用 MATLAB 金融工具箱。

2. 列举 MATLAB 的金融工具箱内容。

3. 探讨未来金融业发展方向，思考 MATLAB 金融工具箱将会起到哪些作用，谈谈如何利用工具箱。

第2章 MATLAB数值计算初步

本章主要介绍MATLAB数值计算基本功能，要求掌握变量与常量基本用法，熟悉单元数组与结构数组的特点；掌握向量与数组基本运算，利用索引函数对变量进行操作；掌握矩阵运算，学会对特殊形式矩阵进行赋值；掌握MATLAB语句用法，规范编程。

2.1 数据类型

MATLAB数据类型主要包括数字、字符串、向量、矩阵、单元数组以及结构数组。下面详细介绍这几种数据类型。

MATLAB中变量命名原则如下：

- 变量名大小写代表不同变量。
- 变量名长度不超过31位，31位后面的字符被忽略。
- 变量名必须以字母开头，变量名中可以有字母、数字、下划线，但不能有标点符号。

在MATLAB中变量分为局部变量与全局变量，全局变量在主程序与函数文件中有效，局部变量仅在函数文件中有效。如果定义全局变量，则需要对变量事先进行声明，即在变量前加上声明global。

在MATLAB中还有一些内部定义变量，例如pi表示圆周率，NaN表示不确定值（根据国际电子和电器行业协会（IEEE）制定的标准，含义是Not a Number），eps表示浮点运算相对精度为10^{-52}，Inf表示无穷大，Realmax表示最大正浮点数2^{1023}，Realmin表示最小正浮点数2^{-1022}。

2.1.1 数字变量

在MATLAB中一般代数表达式可以直接在提示符下演算，非常方便，如四则运算符号可直接用+、-、*、/，因此MATLAB被称为演算纸式科学计算语言。

对于简单数字运算，可以直接在Command窗口中以惯用形式输入，例如计算23×45，可以在窗口直接执行命令：

```
>> 23 * 45
ans =
      1035
```

计算结果自动保存在变量 ans(MATLAB 自动生成变量 ans 保存运算结果)中。如果表达式后面是分号,计算的结果不显示在屏幕上,这样可以明显加快运算速度。

在 MATLAB 中数值有多种显示格式,在默认情况下,如果数据为整数,则以整型表示;如果为实数则保留小数点后面 4 位浮点型。MATLAB 中的所有数据按照 IEEE 浮点标准所规定的长型格式存储,数值有效范围是 $10^{-308} \sim 10^{308}$。

MATLAB 输入格式完全继承了 C 语言风格,如正负号、小数点和科学计数法等。

MATLAB 输出格式可由 format 函数控制。format 函数主要功能是控制变量输出形式,其内容如表 2.1 所列。

表 2.1 format 函数内容

类 型	结 果	例 子
+	+,-,空格	+
bank	美元或者美分,保留小数点后 2 位有效数字	3.14
compact	尽量把输出的形式控制在一行	theta = pi/2 theta=1.5708
hex	十六进制	
long	长型输出,双精度 15 位,单精度 8 位	3.14159265358979
long e	长型输出,浮点格式,双精度 15 位,单精度 8 位	3.141592653589793e+00
long g	最合适的长型输出,双精度 15 位,单精度 8 位	3.14159265358979
loose	换行输出,便于阅读	theta = pi/2 theta=1.5708
rat	以最简分数形式输出	3/2
short	短型输出,保留 5 位有效数字	3.1416
short e	短型浮点输出,保留 5 位有效数字	3.1416e+00
short g	短型最合适输出,保留 5 位有效数字	3.1416

下面以 sqrt(3)为例说明 format 的运用。代码如下:

```
>> a = sqrt(3)
a =
   1.7321
>> format short;a
a =
   1.7321
>> format long;a
a =
   1.73205080756888
>> format hex;a
a =
```

```
  3ffbb67ae8584caa
>> format bank;a
a =
  1.73
>> format + ;a
a =
+
>> format long e;a
a =
   1.732050807568877e+000
>> format short e;a
a =
  1.7321e+000
>> format short g;a
a =
  1.7321
>> format long g;a
a =
  1.73205080756888
>> format rational;a
a =
  1351/780
```

如果查询当前输出格式，可以进行如下操作：

```
>> format long
>> get(0,'Format')
ans =
     long
```

可以知道当前输出形式为 long 型格式输出。

2.1.2 字符串操作

字符串运算是高级语言不可缺少的部分，MATLAB 字符串运算功能是非常丰富的，特别是 MATLAB 增加了符号运算工具箱(Symbolic Math Toolbox)之后，字符串函数运算功能大大加强，成为 MATLAB 符号运算表达式基本单元。

1. 字符串赋值

MATLAB 可以直接对字符串进行赋值。例如：

```
>> a='hello'
a =
   hello
```

这样就把字符串 hello 值赋给了变量 a。

上述过程可以通过调用函数 char 来完成。例如：

```
>> a=char('h','e','l','l','o');
>> a=a'
a =
   hello
```

此外字符串中每个字符(含空格)都是字符串数组一个元素。例如：

```
>> s='i use MATLAB'
s =
   i use MATLAB
```

size 函数用来查看变量维数，s 是一个 1 行 12 列字符串向量：

```
>> size(s)
ans =
     1     12
```

利用矩阵的方式同样可以对字符串变量进行赋值：

```
>> s=['i use MATLAB']
s =
   i use MATLAB
```

或者执行如下操作：

```
>> s='i use MATLAB'
s =
   i use MATLAB
```

2. 字符串和数组之间格式转换

将字符串转换为 ASCII 代码是 double 函数。

```
>> s=['i use MATLAB'];
>> double(s)
ans =
   105    32   117   115   101    32   109    97   116   108    97    98
```

把字符数组转化为字符串是 cellstr 函数：

```
>> s = ['i use MATLAB']';
>> cellstr(s)
ans =
        'i'
        ''
        'u'
        's'
        'e'
        ''
        'm'
        'a'
        't'
        'l'
        'a'
        'b'
```

3. 数值与数字字符串

将数值转换为数字字符串，保持 2 位有效数字。

```
>> B = rand(2,4)
B =
    0.9501    0.6068    0.8913    0.4565
    0.2311    0.4860    0.7621    0.0185
>> B_str = num2str(B,2)
B_str =
        0.95      0.61      0.89      0.46
        0.23      0.49      0.76     0.019
```

将数字字符串转换为数值。

```
>> B = str2num(B_str)
B =
    0.9500    0.6100    0.8900    0.4600
    0.2300    0.4900    0.7600    0.0190
```

4. 把字符串转换为双精度数值

```
>> str2double('23 + 45i')
```

```
ans =
     23.0000 + 45.0000i
```

具体而言，字符串与数组之间转换函数如表 2.2 所列。

表 2.2 字符串与数组之间转换函数

函数名	功能	函数名	功能
num2str	数字转换字符串	str2num	字符串转换数字
int2str	整数转换字符串	str2int	字符串转换为整数
mat3str	矩阵转换为字符	str2mat	字符串转换为矩阵

5. 字符串操作

MATLAB 对字符串操作与 C 语言几乎相同，如表 2.3 所列。

表 2.3 字符串操作函数

函数名	功能	函数名	功能
strcat	连结串	strrep	串替代
strvcat	垂直连接串	strtok	寻找串中的记号
strcmp	串比较	upper	大写串
strncmp	比较串前 n 个字符	lower	小写串
findstr	在其他串中寻找此串	blanks	空格串
strjust	串对齐	deblank	删掉串中空格
strmatch	查找匹配的字符串		

6. 字符串对变量操作

如果已知 a1＝1，a2＝2，a3＝3，a4＝4，a5＝5；现在需要把这 5 个变量的值放在一个数组变量 a 中。在 MATLAB 中编写脚本文件 utility.m 如下：

```
name = 'a';
for i = 1:5
name(2) = int2str(i);                 % 把 name 中第二个字母换成 i
a(i) = eval(name);                    % eval 将 name 变成运算表达式，并进行运算
end
a
```

运行脚本文件可以看到 a 的内容。

```
>> utility
a =
     1     2     3     4     5
```

7. 字符串循环

字符串循环是 MATLAB 中具有特色的功能，通过对字符串中字符做循环实现字符串索引。

下面对字符串 utility 进行循环。

```
s='utlity';
for i=s
    i
end
```

结果显示如下：

```
i =
u
i =
t
i =
l
i =
i
i =
t
i =
y
```

2.1.3　单元变量与结构变量

单元型变量是 MATLAB 中较为特殊的一种数据类型，从本质上讲，单元型变量实际上是一种以任意形式数组为元素的多维数据。

1. 单元型变量的定义

单元矩阵中不同位置可有不同数据类型，单元数组中不同位置也可以有不同数据类型，这就使单元数组与单纯由数字组成数组和单纯由字符串组成的文本数组有着不同特点。单元数组可以包含几个字符串、几个数字。

MATLAB 中有多种方法构建单元数组。单元型变量定义可以通过两种方式定义：一种是用赋值语句直接定义；另外一种是由 cell 函数预先分配存储空间，然后定义。

```
>> a = [1 2;3 4];
>> b = {1:4,a,'i use MATLAB'}
b =
    [1x4 double]    [2x2 double]    'i use MATLAB'
```

如果需要查看每个单元内容可以输入

```
>> b{1}
ans =
     1     2     3     4
>> b{2}
ans =
     1     2
     3     4
>> b{3}
ans =
     i use MATLAB
```

2. 单元型变量操作函数

在 MATLAB 语言中，与单元型变量相关的操作函数如表 2.4 所列。

表 2.4 单元型变量操作函数

函数名	功 能	函数名	功 能
cell	生成单元型变量	deal	输入/输出控制
cellfun	作用于单元变量函数	cell2struct	单元型转化结构型
celldisp	显示单元变量内容	struct2cell	结构型转化为单元型
cellplot	图形显示单元变量	iscell	判断是否为单元型
num2cell	数值型转变为单元型	reshape	改变单元数组结构

下面对单元变量 f 进行赋值。

```
>> f = {'MATLAB',45,78,32}
```

用 cellplot 函数显示 f 的内容，如图 2.1 所示。

```
>> cellplot(f)
```

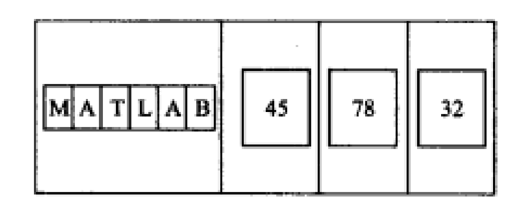


图 2.1　单元型变量内容

3. 结构变量

在 MATLAB 中的结构变量类似于 C 语言中的"结构体"数据，它包含了许多不同类型的数据项，结构数组中不同的数据项即为一个属性。

在 MATLAB 中提供了一些专门对结构数组的操作函数，如表 2.5 所列。

表 2.5　结构数组的操作函数

函数名	功　能	函数名	功　能
fieldname	查询结构数组的属性名	isfield	返回 1 表示数组，0 不是
getfield	获取结构数组的属性值	rmfield	删除属性
isstruct	返回 1 表示是结构数组，0 不是结构数组	setfield	设置属性值

例如，深发展的股价和成交量如表 2.6 所列。

表 2.6　深发展(SZ000001)的股价和成交量

深发展 / 日　期	股价/元	成交量/股
2007-8-20	37.9	36 638 584
2007-8-21	38.56	33 599 021

下面可以用结构创建深发展的股价和成交量。

第一种方法是直接用 struct 函数创建。

```
>> sz000001 = struct('date','2007-8-20','close',37.9,'vol',36638584)
sz000001 =
        date: '2007-8-20'
        close: 37.9000
        vol: 36638584
```

这样就创建了 sz000001 结构变量。

第二种方法是采用赋值语句创建结构数据。

```
>> zs000001.date = '2007 - 8 - 20';
>> zs000001.close = 37.90;
>> zs000001.vol = 36638584
zs000001 =
            date: '2007 - 8 - 20'
            close: 37.9000
            vol: 36638584
```

下面输入下一个交易日的数据。

```
>> zs000001(2).date = '2007 - 8 - 21';
>> zs000001(2).close = 38.56;
>> zs000001(2).vol = 333599021
zs000001 =
            1x2 struct array with fields:
            date
            close
            vol
```

下面开始对 sz000001 中的数据进行访问。

```
>> zs000001(1)
ans =
      date: '2007 - 8 - 20'
      close: 37.9000
      vol: 36638584
>> zs000001(2)
ans =
      date: '2007 - 8 - 21'
      close: 38.5600
      vol: 333599021
```

2.1.4 单元变量与结构变量之间的转换

1. 数组矩阵转换为单元数组

magic 矩阵为魔方矩阵，该矩阵为方阵，且每行每列主对角线上的元素全部相等。

```
>> a = magic(3)
```

```
a =
    8      1      6
    3      5      7
    4      9      2
>> b = num2cell(a)
b =
    [8]    [1]    [6]
    [3]    [5]    [7]
    [4]    [9]    [2]
```

如果把单元数组转换为数组则输入

```
>> c = cell2mat(b)
c =
    8      1      6
    3      5      7
    4      9      2
```

2. 结构数组转换为单元数组

```
>> sz000001 = struct('date','2007-8-20','close',37.9,'vol',36638584)
sz000001 =
          date: '2007-8-20'
          close: 37.9000
          vol: 36638584
>> c = struct2cell(sz000001)
c =
    '2007-8-20'
    [  37.9000]
    [ 36638584]
```

下面是将单元数组转换为结构数据。

```
>> sz = {'2007-8-20',37.9,36638584;'2007-8-21',38.56,333599021}
sz =
     '2007-8-20'     [37.9000]     [ 36638584]
     '2007-8-21'     [38.5600]     [333599021]
>> fields = {'date', 'close', 'vol'};
>> sz000001 = cell2struct(sz, fields, 2)
sz000001 =
```

```
        2x1 struct array with fields:
        date
        close
        vol
```

下面查看变量 s 中每个元素的内容。

```
>> sz000001(1)
ans =
    date: '2007 - 8 - 20'
    close: 37.9000
    vol: 36638584
>> sz000001(2)
ans =
    date: '2007 - 8 - 21'
    close: 38.5600
    vol: 333599021
```

2.2 矩阵及向量运算

2.2.1 矩阵生成

1. 在 Command 窗口直接输入矩阵数据

下面在 Command 窗口输入矩阵数据。

```
>> a=[1 2 3 ;4 5 6;7 8 9]
a =
    1    2    3
    4    5    6
    7    8    9
```

2. 单位矩阵

MATLAB 生成单位矩阵是 eye 函数。eye(n)表示生成 $n\times n$ 阶单位阵，eye(m,n)表示生成 $m\times n$ 阶单位阵。下面给出几种生成单位矩阵方法。

生成 3 阶单位方阵。

```
>> eye(3)
```

```
ans =
     1     0     0
     0     1     0
     0     0     1
```

生成 3 行 4 列单位矩阵。

```
>> eye(3,4)
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
```

此外,在 MATLAB 中还可以生成与给定矩阵阶数相同的单位矩阵。例如,给定矩阵 a,下面生成与 a 相同阶数的单位矩阵。

```
>> a=[1,2,3;4,5,6;7,8,9]
a =
   1     2     3
   4     5     6
   7     8     9
>> eye(size(a))
ans =
     1     0     0
     0     1     0
     0     0     1
```

3. 零矩阵

零矩阵就是所有元素都为 0 的矩阵,其函数是 zeros,用法同 eye 函数。

```
>> zeros(3,4)
ans =
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

4. 壹矩阵

壹矩阵就是所有元素都为 1 的矩阵,其函数是 ones,用法同 eye 函数。

```
>> ones(3,4)
ans =
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

如果生成 5 000×2 000 矩阵，里面每个向量都是 7，若用 7 * ones(5000,2000)命令，这样速度比较慢；而用 repmat(7,5000,2000)速度就快多了，至少缩短一半的时间。

5. 利用 diag 函数创建对角矩阵

在 MATLAB 中用 diag 函数来建立对角矩阵，格式为 diag(X,n)，X 表示对角线元素，n 表示对角矩阵元素位置，$n>0$ 时表示上对角矩阵元素，$n<0$ 时表示下对角矩阵元素。

如果需要创建一个对角线元素为 1 到 5 的对角矩阵，可以执行以下命令：

```
>> a=[1,2,3,4,5];
>> diag(a)
ans =
     1     0     0     0     0
     0     2     0     0     0
     0     0     3     0     0
     0     0     0     4     0
     0     0     0     0     5
```

如果需要建立一个次对角线元素为 a 的对角阵则可以执行如下命令：

```
>> diag(a,-1)
ans =
     0     0     0     0     0     0
     1     0     0     0     0     0
     0     2     0     0     0     0
     0     0     3     0     0     0
     0     0     0     4     0     0
     0     0     0     0     5     0
```

6. 矩阵变向

矩阵变向包括矩阵旋转、左右翻转、上下翻转，分别由函数 rot90、fliplr 和 flipud 实现。函数 flipdim 用来对指定维数进行翻转。

```
rot90(A)          将矩阵 A 逆时针方向旋转 90°
```

```
rot90(A,k)        将矩阵 A 逆时针方向旋转(90 * k)°
fliplr(A)         将 A 左右翻转
flipud(A)         将矩阵 A 上下翻转
flipdim(A,dim)    将矩阵 A 第 dim 维翻转,dim = 1 表示对行翻转,dim = 2 表示列翻转
```

```
>> a = [1 2 3;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> rot90(a)
ans =
     3     6     9
     2     5     8
     1     4     7
>> rot90(a, - 1)
ans =
     7     4     1
     8     5     2
     9     6     3
>> fliplr(a)
ans =
     3     2     1
     6     5     4
     9     8     7
>> flipdim(a,1)
ans =
     7     8     9
     4     5     6
     1     2     3
```

7. 矩阵迹

在 MATLAB 中求矩阵迹函数是 trace。例如,求一个 3 阶魔方矩阵迹,可以执行如下命令:

```
>> a = magic(3)
a =
     8     1     6
     3     5     7
```

```
     4     9     2
>> trace(a)
ans =
      15
```

计算出矩阵迹为 15。

8. 矩阵行列式

在 MATLAB 中计算矩阵行列式的值用函数 det。

求矩阵 $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ 行列式的值。

```
>> a=[1 2 3 ;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> det(a)
ans =
     0
```

9. 矩阵特征值

在 MATLAB 中计算特征值的函数是 eig，下面求矩阵 $\boldsymbol{a}=\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ 特征值。

```
>> a=[1 2 3 ;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> eig(a)
ans =
   16.1168
   -1.1168
   -0.0000
```

矩阵特征值分别为 16.116 8,−1.116 8,0。

10. 取整运算

在 MATLAB 中的取整运算有 ceil,floor,round,fixed 函数。其中 ceil(A)为取大于 A 的整数(又称向上取整),floor (A)为取小于 A 的整数(向下取整),round(A)为四舍五人,fix (A)为直接舍入小数。下面采用不同方法进行取整。

```
>> a = [-1.9, -0.2, 3.4, 5.6, 7];
```

如果向上取整需输入如下命令:

```
>> ceil(a)
ans =
    -1     0     4     6     7
```

如果向下取整需输入如下命令:

```
>> floor(a)
ans =
    -2    -1     3     5     7
```

如果是四舍五人法则需输入如下命令:

```
>> round(a)
ans =
    -2     0     3     6     7
```

如果直接舍入小数部分需输入如下命令:

```
>> fix(a)
ans =
    -1     0     3     5     7
```

金融计算中有时用到取整运算,例如中证指数公司的《沪深 300 指数编制细则》中规定进入沪深 300 成分股采用分级靠档的办法。分级靠档的办法是根据每只股票流通股的比率计算权重,沪深 300 指数分级靠档办法如表 2.7 所列。

表 2.7　沪深 300 指数分级靠档表

项　目	分级指数								
流通比率/(%)	≤10	(10~20]	(20~30]	(30~40]	(40~50]	(50~60]	(60~70]	(70~80]	>80
加权比率/(%)	全部流通股	20	30	40	50	60	70	80	总股本

例如,某股票的流通股占总股本的 7 %,则全部流通股都进入指数,如果流通股占总股本的 35 %,则总股本的 40 %进入指数。例如,2007 年 6 月 18 日,上海、深圳两市 4 家 A 股的总股本与流通股如表 2.8 所列。

表 2.8 国内 4 家 A 股的股本表

股票代码	股票名称	总股本/万股	流通股/万股	流通股比率	折算后比率
000040	许继电气	37 827	26 404	0.698 02	0.7
002024	苏宁电器	144 150	58 840	0.408 186	0.5
000912	泸天化	58 500	26 715	0.456 667	0.5
600108	亚盛集团	144 119	114 554	0.794 857	0.8

计算权重方法如下:

```
>> a=[37827    144150    58500    144119];          % 总股本
>> b=[26404    58840    26715    114554];           % 流通股
>> wei=b./a
wei =
      0.6980    0.4082    0.4567    0.7949
>> wei1=ceil(10*wei)
wei1 =
       7     5     5     8
>> wei1=wei1/10
wei1 =
        0.7000    0.5000    0.5000    0.8000
```

这样各个股票计入指数部分的比例就确定了。

2.2.2 向量运算

在 MATLAB 中矩阵运算的书写习惯和人们日常的书写习惯很相似,但需要注意的是,矩阵运算符前面不用加点号;特别地,点除“./”表示对应元素相除,而向量相除运算其前面需要加点号。

1. 向量点乘

向量点乘表示对应元素相乘。

```
>> a=[1 2 3];b=[2 3 4];
>> a.*b
ans =
```

```
     2     6    12
```

也可以对向量中的每个元素分别进行幂运算。

```
>> a=[1 2 3];b=[2 3 4];
>> a.^b
ans =
       1     8    81
```

如果a,b都是同阶矩阵,则同样表示矩阵中对应元素相乘。

```
>> a=[1,2,3;4 5 6; 7 8 9];b=eye(3);
>> a.*b
ans =
     1     0     0
     0     5     0
     0     0     9
```

表2.9是邯郸钢铁、ST东北高速、恒瑞医药和深达声2007年9月13日的收盘价以及成交量,试计算成交金额。

表2.9 部分股票的价格及成交量

项 目	邯郸钢铁	ST东北高速	恒瑞医药	深达声
收盘价/元	8.56	6.51	42.13	7.99
成交量/股	86 384 700	9 580 900	919 000	2 722 900

```
>> price=[8.56 6.51 42.13 7.99];                    %每只股票价格
>> Vol=[86384700  9580900  919000  2722900];        %每只股票成交量
>> total=price.*Vol                                 %计算每只股票的成交金额
total =
        739453032     62371659     38717470     21755971
```

2. 向量内积计算

向量内积是指两个向量对应元素乘积再求和,两个向量维数应该相同,例如$(a_1,a_2,a_3,\cdots,a_n)$和$(b_1,b_2,b_3,\cdots,b_n)$内积为$a_1b_1+a_2b_2+a_3b_3,\cdots,a_nb_n$。在MATLAB中向量内积的函数为dot。

```
>> a=[1 2 3];b=[2 3 4];
>> dot(a,b)
```

```
ans =

      20
```

下面计算表 2.9 中 4 只股票的总的成交金额。

```
>> price=[8.56 6.51 42.13 7.99];                              % 每只股票价格
>> Vol=[86384700  9580900  919000  2722900];                  % 每只股票成交量
>> total=dot(price,Vol)
total =

        862298132
```

3. 数据索引

find 函数是 MATLAB 中比较常用的查找函数，例如 **X**= [1　0　4　−3　0　0　0　8　6]，如果要使 **X** 中大于 0 的数变成 1，则可以执行以下函数：

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> y=find(X>0);
>> X(y)=1
X=
     1     0     1    -3     0     0     0     1     1
```

或者执行如下命令：

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> X(find(X>0))=1
X =
     1     0     1    -3     0     0     0     1     1
```

进一步地，如果要求 **X** 中大于 0 小于 2 的数变为 1，则

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> X(find(X>0 & X<1))=1
X =
     1     0     4    -3     0     0     0     8     6
```

如果要求 **X** 中前 2 个大于 0 的数变为 1，则

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> X(find(X>0 ,2))=1
X =
     1     0     1    -3     0     0     0     8     6
```

对于矩阵而言，情况变得要复杂一点，例如，求 3 阶魔方矩阵中大于 3 的元素。find 对矩阵中元素进行搜寻时顺序是从第一列开始从上到下，然后从第二列开始从上到下进行计数。

```
>> M = magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
>> r = find(M>3)
r =
     1
     3
     5
     6
     7
     8
>> M(find(M>3)) = 1
M =
     1     1     1
     3     1     1
     1     1     2
```

实际上借助逻辑函数 abs 可以实现类似功能。下面将魔方矩阵中大于 3 的元素改为 2。

```
>> a = magic(3)
a =
     8     1     6
     3     5     7
     4     9     2
>> L = abs(a>3)
L =
     1     0     1
     0     1     1
     1     1     0
>> a(L) = 2
a =
     2     1     2
     3     2     2
     2     2     2
```

4. 查找矩阵中不相同元素个数

在矩阵计算中有时需要统计不相同元素个数，可以用 unique 函数。

```
>> a = 1:9
a =
    1     2     3     4     5     6     7     8     9
>> unique(a)
ans =
    1     2     3     4     5     6     7     8     9
```

表示不相同元素个数为 9 个。下面是一个例子。

```
>> a = [1 2 2 4 4 5]
a =
    1     2     2     4     4     5
>> unique(a)
ans =
       1     2     4     5
```

表示不相同元素个数有 4 个，分别是 1,2,4,5。

```
>> length(unique(a))
ans =
       9
```

其中 unique(a)找出不相同元素位置，length 计算出 ***a*** 中元素个数。

5. 极大值

调用方式

```
C = max(A)
C = max(A,[],dim)
```

输入参数

A	向量或者矩阵
dim	dim = 1(默认值)时分别求每列极大值;dim = 2 时分别求每行极大值

下面是一个例子。

```
>> x = [1 2 3; 4 5 6]
x =
```

```
     1     2     3
     4     5     6
>> max(x,[],2)
ans =
     3
     6
>> max(x,[],1)
ans =
     4     5     6
```

2.2.3 矩阵运算

1. 矩阵的运算

1) 矩阵累积求和

调用方式

```
a = cumsum(x,n)
```

输入参数

x	矩阵
n	$n=1$,表示沿行方向累积求和;$n=2$,表示沿列方向累积求和

例如,对矩阵沿列方向累积求和。

```
>> a = ones(2,5)
a =
     1     1     1     1     1
     1     1     1     1     1
>> cumsum(a,2)
ans =
     1     2     3     4     5
     1     2     3     4     5
```

对矩阵沿行方向累积求和。

```
>> cumsum(a,1)
ans =
     1     1     1
     2     2     2
```

```
     3     3     3
```

2) 矩阵连乘

调用方式

```
B = prod(x)
B = prod(x,n)
```

输入参数

x	矩阵
n	$n=1$，表示沿行方向连乘；$n=2$，表示沿列方向连乘

例如，对向量每一个元素连乘。

```
>> a=[1,2,3,4];
>> prod(a)
ans =
      24
```

下面是矩阵连乘。

```
>> a=[1 2;3 4]
a =
     1     2
     3     4
>> prod(a,1)                    %行方向连乘
ans =
     3     8
>> prod(a,2)                    %列方向连乘
ans =
     2
    12
```

3) 矩阵累积连乘

调用方式

```
B = cumprod(x)
B = cumprod(x,dim)
```

输入参数同上。

下面是矩阵累计连乘的例子。

```
>> a=[1 2;3 4]
```

```
a =
    1    2
    3    4
>> cumprod(a,1)                % 行方向累积连乘
ans =
     1    2
     3    8
>> cumprod(a,2)                % 列方向累积连乘
ans =
     1    2
     3   12
```

2. MATLAB 中数据显示格式

MATLAB 中有两种方式显示数据格式：一是通过 MATLAB 界面上菜单进行设置；二是在 MATLAB 的 Command 窗口中直接执行相关命令。用菜单设置 MATLAB 数据显示格式时，单击 File 菜单的 Preference 选项，将会出现 Data 对话框，这时可以借助于滚动条来设置，除了可以选择数据显示位数长短外，还可以设置数据显示格式是疏松型(loose)还是紧凑型(compact)。

3. LU 分解

高斯消元法求解线性方程，包括把增广矩阵转换为三角矩阵形式的过程，消去阶段工作步骤是把矩阵 ***A*** 分解成为下三角矩阵 ***L*** 与上三角矩阵 ***U*** 的乘积。这种计算矩阵 ***L***、***U*** 的过程称为 LU 分解法。如果矩阵 ***A*** 是正定矩阵，那么进一步地还有更高效的 Cholesky 分解法。$A=C'C$，其中 C 是上三角矩阵。

调用方式

```
[L,U] = lu(X)
Y = lu(X)
[L,U,P] = lu(X)
[L,U,P,Q] = lu(X)
[L,U,P] = lu(X,thresh)
[L,U,P,Q] = lu(X,thresh)
```

在[L,U] = lu(***X***)中，***L*** 为置换矩阵，***U*** 为下三角矩阵，满足 $\boldsymbol{L}*\boldsymbol{U}=\boldsymbol{X}$。

[L,U,P] = lu(***X***)中 ***L*** 为上三角矩阵，***U*** 为下三角矩阵，***P*** 为置换矩阵，满足 $\boldsymbol{L}*\boldsymbol{U}=\boldsymbol{P}*\boldsymbol{X}$。

下面给出矩阵 ***A*** 形式，对矩阵 ***A*** 进行 ***LU*** 分解，代码如下：

```
>> A = [1 2 3;4 5 6;7 8 0];
>> [L,U] = lu(A)
L =
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
    1.0000         0         0
U =
    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0     4.500
```

给出矩阵 $\boldsymbol{L}$ 与 $\boldsymbol{U}$ 形式满足要求。

如果 $\boldsymbol{L}$ 为下对角矩阵,主对角线上元素为 1,$\boldsymbol{U}$ 为上对角矩阵。

```
>> [L,U,p] = lu(A)
L =
    1.0000         0         0
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
U =
    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000
p =
     0     0     1
     1     0     0
     0     1     0
```

下面验证 $\boldsymbol{L}*\boldsymbol{U}=\boldsymbol{p}*\boldsymbol{A}$

```
>> L*U-p*A
ans =
     0     0     0
     0     0     0
     0     0     0
```

如解线性方程组

$$\begin{cases} x_1+0.2x_2+0.5x_3=1 \\ 0.2x_1+0.4x_2+x_3=2 \\ 0.5x_1+0.1x_2+06x_3=1.5 \end{cases}$$

记

$$\boldsymbol{A}=\begin{pmatrix} 1 & 0.2 & 0.5 \\ 0.2 & 0.4 & 1 \\ 0.5 & 0.1 & 0.6 \end{pmatrix},\boldsymbol{x}=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},\boldsymbol{b}=\begin{pmatrix} 2 \\ 2 \\ 1.5 \end{pmatrix}$$

则线性方程组可写为 $\boldsymbol{Ax}=\boldsymbol{b}$。

根据 LU 分解有 $\boldsymbol{LUx}=\boldsymbol{b}$，记 $\boldsymbol{Y}=\boldsymbol{L}^{-1}\boldsymbol{b}$，则 $\boldsymbol{x}=\boldsymbol{U}^{-1}\boldsymbol{Y}$ 为方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 解。

```
>> a = [1 0.2 0.5;0.2 0.4 1;0.5 0.1 0.6];
>> b = [1 2 1.5]';
>> [L,U] = lu(a)
L =
    1.0000         0         0
    0.2000    1.0000         0
    0.5000         0    1.0000
U =
    1.0000    0.2000    0.5000
         0    0.3600    0.9000
         0         0    0.3500
>> y = L^-1 * b;
>> x = U^-1 * y;
```

从上面结果可以看出，$\boldsymbol{L}$ 为下对角矩阵，$\boldsymbol{U}$ 为上对角矩阵，$\boldsymbol{x}$ 是方程解。

4. 消去单一维矩阵

在多维矩阵中，如果有一个是一维，这个一维是多余的，可以消去。

调用方式

```
B = squeeze(A)
```

下面构建一个 2×1×2 矩阵。

```
>> price(1,1,1) = 1;
>> price(1,1,2) = 2;
>> price(2,1,1) = 3;
>> price(2,1,2) = 4
```

```
price(:,:,1) =
              1
              3
price(:,:,2) =
              2
              4
```

然后消除中间的维数。

```
>> B = squeeze(price)
B =
    1     2
    3     4
```

2.2.4 排　序

MATLAB对矩阵排序函数是sort，其调用方式如下：

```
B = sort(x)
B = sort(x,n)
[B,ind] = sort(x,n ,mode)
```

输入参数

x	矩阵
n	$n=1$表示沿行方向排序；$n=2$表示沿列方向排序
mode	mode = 'descend'表示按照递减排序，mode = 'ascend'表示按照递增排序

输出参数

B	排序后的矩阵
ind	B中元素在x中的位置

下面是一个例子。

```
>> a = [ 3 7 5; 0 4 2 ];
>> sort(a,1)
ans =
     0     4     2
     3     7     5
>> sort(a,2)
ans =
```

```
     3     5     7
     0     2     4
>> [B,ind] = sort(a,2)
B =
     3     5     7
     0     2     4
ind =
     1     3     2
     1     3     2
```

如果按照降序排列可以将 mode 设为'descend'。

```
>> [B,ind] = sort(a,2,'descend')
B =
     7     5     3
     4     2     0
ind =
     2     3     1
     2     3     1
```

下面是用 sort 函数对字母排序的一个例子。

```
>> A = ['one  ';'two  ';'three';'four ';'five ']
A =
    one
    two
    three
    four
    five
>> sort(A,1)
ans =
      fhe
      fio
      onre
      toue
      twvre
```

如果对字母按照字典方式排序可以用 sortrows 函数。

```
>> sortrows(A)
```

```
ans =
      five
      four
      one
      three
      two
```

2.3 插　值

在 MATLAB 中涉及的插值函数主要有一维插值(interp1)、二维插值(interp2)、三维插值(interp3)和样条插值(spline)函数。

2.3.1 一维插值

如果给出一组数组 x,y,则需要找出 x,y 之间的函数关系 $f()$,例如 $y=f(x)$,然后给定 $x1$,求 $y1$,使得 $y1=f(x1)$,这样函数关系 f 有很多种,分别称作不同插值方法。如果 x 是一维变量就称为一维插值;如果 x 是二维变量就称为二维插值;类似地有三维插值。

调用方式

```
Y1 = interp1(x,y,x1,'method')
```

输入参数

x	自变量
y	因变量
x1	给定的自变量值
method	插值算法,默认值为线性插值。其取值如表 2.10 所列

输出参数

Y1	与自变量 $x1$ 对应函数值

表 2.10　插值算法特点比较

插值方法	含　义	特　点	用　途
linear	线性插值	较快	最常用
cubic	三次多项式插值	较慢,精度高	作平滑用
spline	三次样条插值	最慢,精度高,平滑	作平滑用
nearst	最邻近插值	最快,精度低	特大数据处理

注:函数 interp2、interp3 中对 method 的赋值同 interp1

例如，自变量保存在数组 x 中，与 x 对应值保存在 y 中，求 $x1=0.34$ 时对应的 y 值。

```
>> x = 0:0.1:2;y = 2 * x;
>> x1 = 0.34;
>> y1 = interp1(x,y,x1,'linear')                        % 线性插值
y1 =
   0.6800
>> y1 = interp1(x,y,x1,'spline')                        % 样条插值
y1 =
    0.6800
> y1 = interp1(x,y,x1,'cubic')                          % 三次插值
y1 =
    0.6800
>> y1 = interp1(x,y,x1,'nearest')                       % 最邻近插值法
y1 =
    0.6000
```

从上面计算结果看，最邻近法计算效果最差。

2.3.2 样条插值

插值来源于实际，同时又广泛应用于实际，是一种重要的方法。下面主要讨论线性插值、三次样条插值和 Hermite 插值。

在 MATLAB 中样条函数是 spline，其调用方式如下：

```
pp = spline(x,Y)
yy = spline(x,Y,xx)
```

注意自变量 x 是单调增加的。下面是一个样条函数插值例子。

```
>> x = 0:0.1:2;y = 2 * x;
>> x1 = 0.34;
>> y1 = spline(x,y,x1)
Y1 =
    0.6800
```

2.3.3 Hermite 插值

很多实际问题不但要求在节点上相等，而且节点处导数也相等，甚至高阶导数也相等。

Hermite 插值可以保持节点处导数相一致。为简单起见，下面讨论 3 阶 Hermite 插值，相应函数为 pchip。

```
>> x = -3:3;
>> y=2*x;
>> t=-1:0.5:1;
>> p = pchip(x,y,t)              % x 为自变量,y 为因变量,t 为待插值向量
p =
    -2    -1    0    1    2
```

2.4 数值拟合

2.4.1 最小二乘拟合

在数值计算中，有时观察数据存在误差，所以并不要求 $y=f(x)$ 经过所有的样本点 (x_i, y_i)，而只要求在给定点 x_i 上误差 $\delta=f(x_i)-y_i$ 的误差最小，通常使用最小二乘法。

MATLAB 最小二乘拟合方法有多项式拟合，函数名称为 polyfit。下面是一个拟合利率的例子。表 2.11 是日期和利率水平。

表 2.11　日期和利率水平

日　期	1	2	3	4	5
利　率	0.018 7	0.023 8	0.053 3	0.064 0	0.082 5

```
>> x=1:5;
>> y=[0.0350    0.0361    0.0335    0.0682    0.0678];
>> para=polyfit(x,y,2)          % 2 表示二次多项式
para =
    0.0006    0.0133    0.0021
```

这样可以知道拟合的二次曲线为 $y=0.0021+0.0133x+0.0006x^2$。

```
%画出利率图和拟合曲线
>> y1 = 0.0021 + 0.0133 * x1 + 0.0006 * x1.^2;
>> plot(x,y,'b*',x1,y1,'b-')
```

图 2.2 是二次多项式拟合利率图。

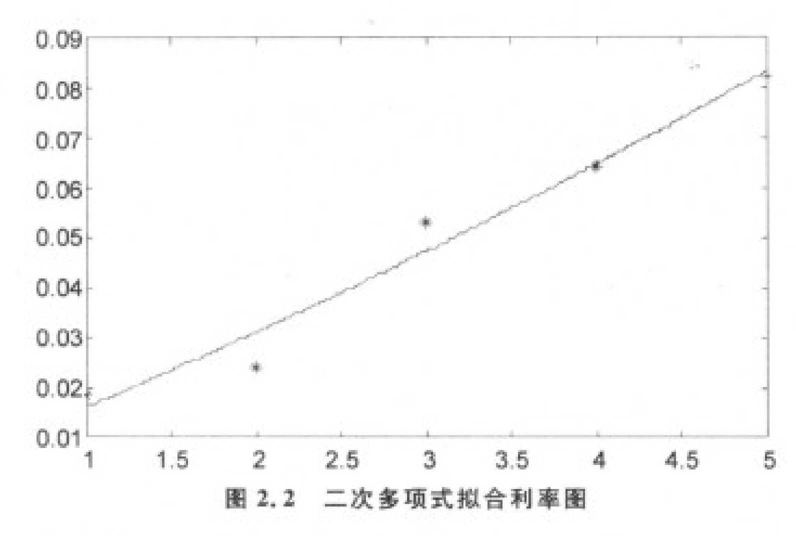


图 2.2　二次多项式拟合利率图

2.4.2　拟合工具箱

首先绘制散点图。

```
>> plot(x,y,'r*')
```

在菜单中选择【Basic Fitting】菜单命令，如图 2.3 所示。

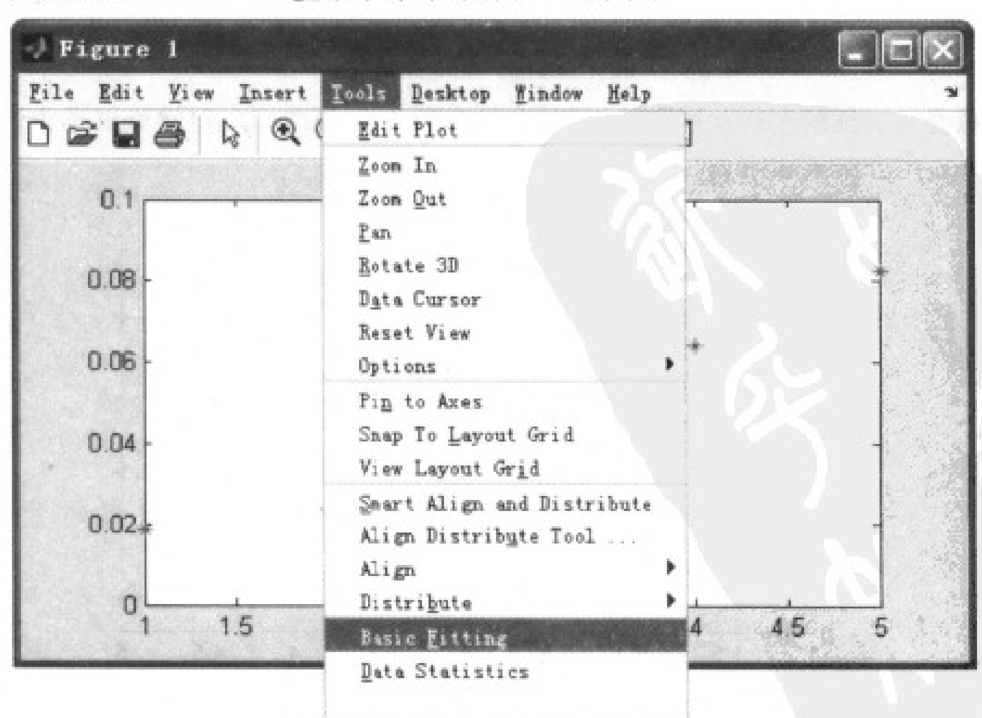


图 2.3　基本拟合菜单

在选择【Basic Fitting】菜单项后系统弹出对话框，如图 2.4 所示。

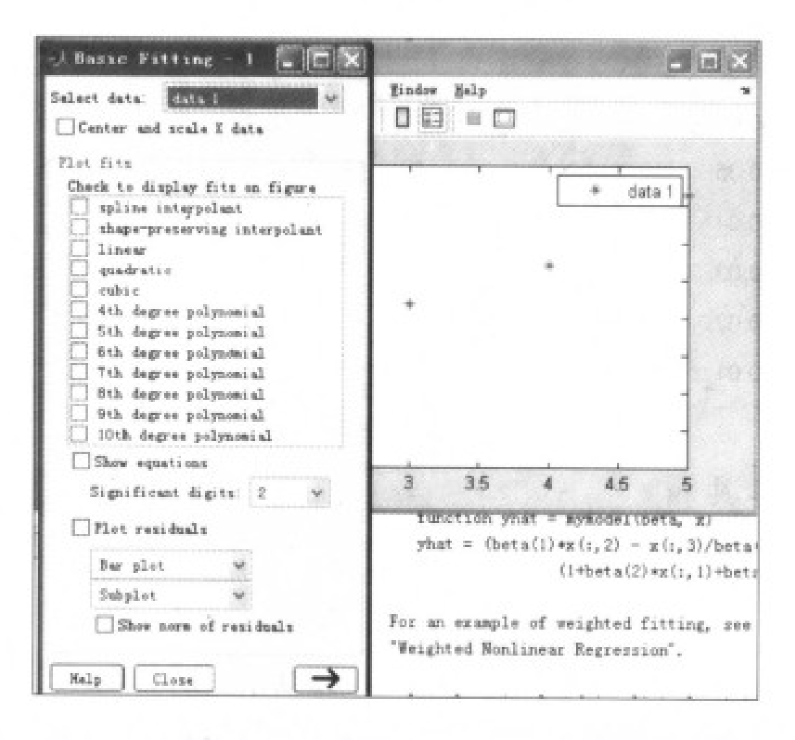


图 2.4　基本拟合对话框

基本拟合对话框的左边是拟合的方法，在拟合的方法前面打个勾就可以选中该方法。例如选中 cubic，拟合结果如图 2.5 所示。

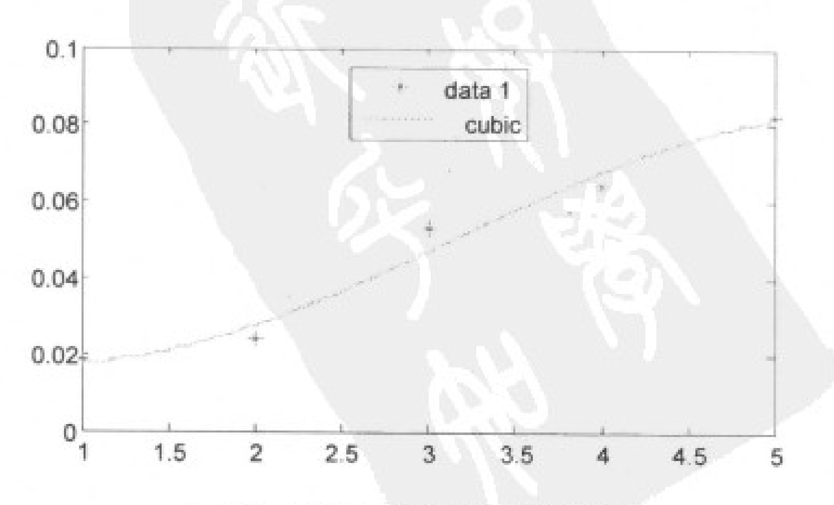


图 2.5　拟合工具菜单中的三次拟合图

2.5　符号计算

MATLAB 不仅可以进行数值计算，而且还可以进行符号计算。

1. 声明符号变量和符号表达式

例如在提示符下输入 3/5，可以得到：

```
>> 3/5
ans =
    0.6000
```

3/5 变成了小数。而用 sym 函数则结果如下：

```
>> sym(3)/sym(5)
ans =
    3/5
```

结果仍然是分数。
比如定义一个符号矩阵：

$$A=\begin{pmatrix}1/2 & 2/3\\ 1/4 & 4/7\end{pmatrix}$$

代码如下：

```
>> A = sym('[1/3,2/3;1/4,4/7]')
A =
    [ 1/3, 2/3]
    [ 1/4, 4/7]
```

2. 合并同类项

用 collect 函数可以合并同类项，例如合并 $x^2+2xy+y^2$ 同类项，代码如下：

```
>> collect(x^2 + 2 * x * y + x * y^2)
ans =
      x^2 + (2 * y + y^2) * x
```

3. 简化符号运算表达式

有时为了计算方便需要对计算结果给出简单的表达式。下面用 simple 函数对公式进行简化。

```
>> syms x y
>> y = x^2 + 2 * x * y + y^2
y =
    x^2 + 2 * x * y + y^2
>> simple(y)
simplify:
x^2 + 2 * x * y + y^2
radsimp:
x^2 + 2 * x * y + y^2
combine(trig):
x^2 + 2 * x * y + y^2
factor:
(x + y)^2
expand:
x^2 + 2 * x * y + y^2
combine:
x^2 + 2 * x * y + y^2
convert(exp):
x^2 + 2 * x * y + y^2
convert(sincos):
x^2 + 2 * x * y + y^2
convert(tan):
x^2 + 2 * x * y + y^2
collect(x):
x^2 + 2 * x * y + y^2
mwcos2sin:
x^2 + 2 * x * y + y^2
ans =
    (x + y)^2
```

公式 $x^2+2xy+y^2$ 简化结果为$(x+y)^2$。

4. 符号矩阵运算

MATLAB 可以完成矩阵四则运算。下面是对符号矩阵的操作。

```
>> syms a b c d x y z
>> A = [a b ;c d]
A =
```

```
    [ a, b]
    [ c, d]
>> B = sym('[x y ;c d]')
B =
    [ x, y]
    [ c, d]
>> A + B
ans =
      [ a + x, b + y]
      [ 2 * c, 2 * d]
>> A * B
ans =
      [ a * x + b * c, a * y + b * d]
      [ c * x + d * c, c * y + d^2]
```

符号矩阵除法。

```
>> C1 = A/B
C1 =
    [   (- b * c + a * d)/(x * d - c * y), - 1/(x * d - c * y) * (a * y - x * b)]
    [                        0,                           1]
>> C2 = A * inv(B)
C2 =
    [  a * d/(x * d - c * y) - b * c/(x * d - c * y), - a * y/(x * d - c * y) + b * x/(x * d - c * y)]
    [                                         0, - c * y/(x * d - c * y) + d * x/(x * d - c * y)]
```

现在对 $C1$ 与 $C2$ 进行简化，结果如下：

```
>> a1 = simple(C1);
>> a2 = simple(C2);
>> a1
a1 =
    [ (- c * b + a * d)/(x * d - c * y), (- y * a + x * b)/(x * d - c * y)]
    [                    0,                        1]
>> a2
a2 =
    [ (- c * b + a * d)/(x * d - c * y), (- y * a + x * b)/(x * d - c * y)]
    [                    0,                        1]
```

经过简化处理后二者相等。如果需要清除这些变量，可以使用 clear 命令。

```
>> clear a1 a2
```

这样 $a1$,$a2$ 从内存中删掉了。如果全部清除直接用 clear 命令即可。

```
>> clear
```

5. 求符号矩阵秩

```
>> A = sym('[a b ;c d]')
A =
    [ a, b ]
    [ c, d ]
>> rank(A)
ans =
      2
```

矩阵 **A** 的秩为 2。

6. 符号函数求导

例如,对 $y=x^2$ 求导。

```
>> y = sym(x^2)
y =
    x^2
>> diff(y)
ans =
      2 * x
```

例如,$y=x^2\sin t$,求$\frac{\mathrm{d}y}{\mathrm{d}t}$。

```
>> y = sym('x^2 * sin(t)')
y =
      x^2 * sin(t)
>> diff(y,'t')
ans =
      x^2 * cos(t)
```

7. 符号代数方程求解

求下列线性方程组的解。

$$\begin{cases}10x_1 - x_2 = 9\\ -x_1 + 10x_2 - 2x_3 = 7\\ -2x_2 + 10x_3 = 6\end{cases}$$

首先，将上面线性方程组写成矩阵形式。

$$\begin{pmatrix}10 & -1 & 0\\ -1 & 10 & -2\\ 0 & -2 & 10\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\begin{pmatrix}9\\ 7\\ 6\end{pmatrix}$$

然后，利用 MATLAB 中矩阵形式求解。

```
>> a=[10, -1, 0;-1, 10, -2;0, -2, 10]
a =
    10      -1       0
    -1      10      -2
     0      -2      10
>> b=[9;7;6]
b =
     9
     7
     6
>> linsolve(a,b)
ans =
       0.9958
       0.9579
       0.7916
```

上面解是近似解，下面用符号函数求解线性方程 $\begin{cases}x+y=1\\ x-y=0\end{cases}$。

```
>> s=solve('x+y=1','x-y=0','x','y')
s =
    x: [1x1 sym]
    y: [1x1 sym]
```

方程的解被保存在结构数组 s 中，$s.x$ 为 x 的解，$s.y$ 为 y 的解，查看内容如下：

```
>> s.x
ans =
       1/2
>> s.y
```

```
ans =
      1/2
```

方程组的解为 $x=1/2$，$y=1/2$。

8. 非线性方程组求解

例如，求解非线性方程组 $\begin{cases} x+e^y=2 \\ y+e^x=2 \end{cases}$ 的代码如下：

```
>> s = solve('x + exp(y) = 2','y + exp(x) = 2','x','y')
s =
    x: [1x1 sym]
    y: [1x1 sym]
>> s.x
ans =
      1.9952
>> s.y
ans =
      - 5.3542
```

方程组解为 1.995 2 和－5.354 2。

9. 因式分解

在 MATLAB 中能进行因式分解的函数是 factor 函数，包括对整数进行素数分解。

```
>> syms x
>> factor(x^3 + 1)
ans =
      (x + 1) * (x^2 - x + 1)
```

下面是整数素数分解。

```
>> factor(3887)
ans =
      13    13    23
```

从上面的结果可以看出，3 887 可分解为 $3\,887=13\times13\times23$。

2.6 字符串命令

为提高计算的灵活性，MATLAB 提供了专门的把字符串转换为函数运算的功能，可以在

运算中改变执行的指令，实现比较复杂的计算。

2.6.1　计算字符串的值

调用方式1

y = eval('字符串')

调用方式2

y = eval('字符串1','字符串2')

先执行字符串1的值，如果出错则执行字符串2的值。

下面先产生服从正态分布的4阶随机矩阵，然后计算收益率。

```
>> randn('seed',0)
>> eval('t = 4,price = randn(1,t),ret = price(2:t)./price(1:t - 1) - 1')
t =
    4
price =
      1.1650      0.6268      0.0751      0.3516
ret =
      - 0.4619     - 0.8802      3.6831
```

也可以先定义字符串，然后进行串演计算。例如：

```
>> randn('seed',0)
>> str = 't = 4,price = randn(1,t),ret = price(2:t)./price(1:t - 1) - 1'
str =
      t = 4,price = randn(1,t),ret = price(2:t)./price(1:t - 1) - 1
>> eval(str)
t =
    4
price =
      1.1650      0.6268      0.0751      0.3516
ret =
      - 0.4619     - 0.8802      3.6831
```

二者计算的结果相同。串演函数可以和字符串函数结合起来生成更复杂的计算。例如，首先生成一组随机数。

```
>> randn('seed',0)
>> price = randn(1,4)
```

```
price =
       1.1650      0.6268      0.0751      0.3516
```

下面查看变量 price 中第二个元素的内容。

```
>> i = 2,eval(['price(',num2str(i),')']),            % num2str 函数将数值转换为字符串
i =
     2
ans =
      0.626 8
```

2.6.2 函数形式调用

在 MATLAB 中以函数形式调用的是 feval 函数。

调用方式

```
[y1, y2, ...] = feval(@fun, x1, ..., xn)
[y1, y2, ...] = feval('fun', x1, ..., xn)
```

输入参数

@fun	句柄函数
fun	函数文件
x1, ..., xn	函数的输入参数

上述运算等价于 fun(x1, ..., xn)。下面是一个计算 3 阶魔方矩阵特征根的例子。

```
>> A = magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
>> feval(@eig, A)
ans =
       15.0000
        4.8990
       -4.8990
```

下面是以函数文件形式调用。

```
>> feval('eig', A)
ans =
       15.0000
```

```
   4.8990
  -4.8990
```

2.6.3 内联函数

内联函数是MATLAB提供的一个对象,其功能和函数文件一样。

调用方式

```
y = inline('表达式')
```

下面是几个例子。

```
>> f = inline('x * y')
f =
    Inline function:
    f(x,y) = x * y
>> z = f(2,3)
z =
    6
```

下面是一个变量为数组的例子。

```
>> f = inline('x(1) * x(2)')
f =
    Inline function:
    f(x) = x(1) * x(2)
>> x = 3;4
x =
    3      4
>> f(x)
ans =
    12
```

实际上也可以用feval函数实现上述目的。

```
>> feval(f,x)
ans =
      12
```

2.7 逻辑运算

2.7.1 基本逻辑运算

逻辑运算是指大小比较、逻辑与及逻辑非关系，在 MATLAB 中的运算如表 2.12 所列。

表 2.12 逻辑运算符号表

符号运算	功　能	函数名
==	等于	eq
~=	不等于	ne
<	小于	lt
>	大于	gt
<=	小于等于	le
>=	大于等于	ge
&	逻辑与	and
|	逻辑或	or
~	逻辑非	not

下面是几个例子。

```
>> a=[1 2 3; 4 5 6]
a =
     1     2     3
     4     5     6
>> b=[2 1 3; 3 5 2]
b =
     2     1     3
     3     5     2
>> c=a<=b
c =
     1     0     1
     0     1     0
```

矩阵 *c* 保存 ***a***,***b*** 比较的结果。如果矩阵 ***a***,***b*** 对应元素满足判断条件则为 1,不满足则为 0。

2.7.2 逻辑关系函数

逻辑关系的函数运算中，大部分函数是 MATLAB 所特有的，它们给用户带来了很大的便

利。主要逻辑关系函数如表 2.13 所列。

表 2.13　逻辑关系函数

函数名	使用说明	函数名	使用说明
any	任一满足条件返回 1	islogical	是逻辑型数据返回 1
all	所有满足条件返回 1	isletter	是字母返回 1
xor	逻辑或非	any	任一有非 0 元素返回 1
isempty	如果空，则返回 1	isnan	是"NaN"取 1
isequal	相等元素位置返回 1	isinf	是无穷大型数据返回 1
isnumber	是数字型数据返回 1	isfinite	是有限大数据返回 1

2.8　控制语句

2.8.1　for 循环语句

for 循环语句的语法为

```
for x = arry
        MATLAB 语句
end
```

其中 x 称为循环变量，循环执行的次数由 arry 中列的数目决定。下面是几个例子。

对变量 x 进行循环赋值。

```
for i = 1:10
x(i) = i
end
```

也可以用向量赋值方法。

```
x = 1:10
```

可以看出二者的结果是相同的。

下面是一个矩阵循环的例子。

```
>>  x = magic(3)
x =
     8      1      6
     3      5      7
```

```
     4     9     2
>> for k = x,k,end
k =
     8
     3
     4
k =
     1
     5
     9
k =
     6
     7
     2
```

可以看出 k 的取值分别为矩阵 x 中的每列元素。

2.8.2 while 条件循环语句

while 也是循环语句，但是需要进行逻辑判断，语句的语法是

while 判断语句

  MATLAB 语句

end

下面是一个例子，计算 1+2+3+4+5。

```
t = 1;
sum = 0;
while t< = 5
sum = sum + t;
t = t + 1;
end
sum
```

运行结果为

```
sum =
     15
```

2.8.3 if - else - end 条件判断

if - else - end 语句为程序流提供了一种条件判断，当判断的表达式为真时执行命令，否则

不执行命令。

if 判断语句
　　MATLAB 语句
elseif 判断语句
　　MATLAB 语句
end

下面是一个例子。

```
a = 10
if a<8
disp('a<8')
elseif a<9
disp('8<= a <9')
else
disp('a>=9')
end
```

运行结果为

```
a>=9
```

2.8.4　switch－case 语句

switch－case 语句语法为

switch x
case text1
MATLAB 语句
case text2
MATLAB 语句
otherwise
end

下面是一个例子。

```
x = 1
switch x
case 1
disp('x = 1')
otherwise
disp('x~= 1')
```



```
end
```

运行的结果为

```
x = 1
```

如果 x=2,运行结果为

```
x~ = 1
```

2.9 MATLAB编程的基本知识

2.9.1 脚本文件与函数文件

如果完成一件比较复杂的任务,需要输入很多命令,可以把这些命令集成到一个文件,每次只要在 MATLAB 中输入文件名就可以了。这时编写脚本文件最合适,脚本文件运行后,产生的所有变量都保存在 MATLAB 基本工作区中。

函数文件是特殊 M 文件,其常见格式如下:

function 输出变量列表=函数名(输入变量列表)

函数文件相当于对 MATLAB 进行了二次开发,其作用和形式与其他高级语言相同,都是为满足特定目标而编写子函数。函数文件头标志是 function,脚本文件无此要求;脚本文件执行完后,变量仍然保存在内存中,函数文件执行完后变量就被清除,除非对其进行全局变量声明。

2.9.2 P代码文件

当 M 文件首次运行时,MATLAB 首先应该对 M 文件进行语法分析,生成内部准代码文件(Psedocode,简称 P 代码文件)。当下次运行时直接运行内存中的 P 文件,这样就省去了对源文件的语法分析,提高了运行速度。P 文件和 M 文件名是相同的,但扩展名为".P"。

在 MATLAB 环境中,假如存在同名的 M 文件和 P 文件,当该文件被调用时,应首先调用 P 文件。

在 MATLAB 中用 inmem 命令可以列出内存中的所有 P 文件、用 clear 命令可清除内存中的所有 P 文件。

2.9.3 编程注意事项

由于 MATLAB 是解释性语句,运行速度较 VC++、Fortran 等语言要慢得多,因此在编写一些大型程序时需要注意程序的运行速度,尽量减少循环判断等语句的使用;尽量用 MATLAB 自带的函数编写程序。为加快 MATLAB 的运行速度,编写程序时应该注意以下内容:

- 运用矩阵运算，避免使用循环语句。因为MATLAB语言是一种解释性语言，循环语句运算耗费很多时间，因此计算中尽可能地少用循环语句，多采用矩阵运算。如果需要进行多重循环，则在循环的外环执行循环次数少的语句，在内环执行循环次数多的语句。
- 大型矩阵预先定维。对于变量事先赋值，用以确定矩阵维数。例如，$m\times n$ 维矩阵变量 $\boldsymbol{M}$，可以用 zeros(m,n)对 $\boldsymbol{M}$ 进行赋值。
- 优先考虑内部函数。在矩阵运算时应该尽可能采用内部函数，这些函数是由更底层的C语言编写的，运行的速度要快于矩阵的循环运算。
- 优化算法。在近似计算时需要考虑步长、收敛性等问题。采用优化算法不但减少工作量，而且可以提高精度。
- 不要设计多用途的函数。函数集多功能于一身，很可能使函数的理解、测试及维护等变得困难。
- 防止程序中的垃圾代码。程序中的垃圾代码不仅占用额外的内存空间，而且还常常影响程序的功能与性能，很可能给程序的测试、维护等造成不必要的麻烦。
- 防止输入错误。此类错误一般是由于把“＜＝”误写成“＜”，或把“＞＝”误写成“＞”等造成的。由此引起的后果，很多情况下是很严重的，所以编程时，一定要在这些地方小心，当编完程序后，应对这些操作符进行彻底的检查。

2.9.4　程序的调试

1. Echo命令

函数文件在执行时，后面带分号的语句将不会显示出来，这在检查程序时会带来不便。使用echo命令可以将每一步的命令都显示在屏幕上，但是这样会导致程序运行速度减慢很多。echo命令的内容如表2.14所列。

表2.14　echo调试命令

命令名称	功　能
echo on	启动echo功能
echo off	关掉echo功能
echo file on	文件名为file在执行时被显示出来
echo file off	关掉文件file执行时显示的内容
echo file	显示文件echo的状态
echo on all	对所有文件启动echo功能
echo off all	关闭所有文件的echo功能

2. 接受用户键盘输入

函数 input 命令可以用来接受用户的键盘输入。

```
>> n = input(['请输入运行次数',blanks(3)])
```

请输入运行次数　　3

```
n =
     3
```

3. 暂停语句

暂停语句允许暂时停止正在执行的 MATLAB 命令，pause(n)表示暂停执行程序 n 秒。

4. 输入与输出控制

MATLAB 中输入与输出控制格式分自由化(disp)与格式化 (fprintf)两种。自由化格式输出函数是 disp，基本用法如下：

```
>> a = 20;
>> disp(a)
    20
```

disp 允许显示字符型变量。

```
>> disp('i like using MATLAB ')
```

显示结果为字符串。

```
i like using MATLAB
```

而指令 fprintf 则是用来控制输出数据及文字格式，它的基本格式如下：

```
>> fprintf('The area is %8.5f\n', area)
```

两个单引号间包括输出字串 The area is；接着是输出数据格式%8.5f；再来是换行符号以避免下一个输出数据或是提示符号挤在同一行；最后输入要输出的数据名 area。

下面是一个控制输出的例子。

```
>> fprintf('The area is %8.5f\n', area)          % 注意输出格式前须有 % 符号，换行符号需有\符号
   The area is 12.56637                          % 输出值为 8 位数，含 5 位小数
>> fprintf('f_form: %12.5f\n',12345.2)           % 输出值为 12 位数，含 5 位小数
   f_form: 12345.20000
>> fprintf('f_form: %12.3f\n',1.23452)           % 输出值为 12 位数，含 3 位小数
```

```
    f_form: 1.235
>> fprintf('e_form: %12.5e\n',12345.2)        %输出值为指数格式的 12 位数,含 5 位小数
    e_form: 1.23452e+004
>> fprintf('f_form: %12.0f\n',12345.2)        %输出值为整数格式的 12 位数
    f_form: 12345
```

2.9.5 MATLAB 其他常用命令

1. 使用 edit 修改文件

例如查看 magic 函数的内容,可以执行如下命令:

```
>> edit magic
function M = magic(n)
%MAGIC  Magic square.
%   MAGIC(N) is an N-by-N matrix constructed from the integers
%   1 through N^2 with equal row, column, and diagonal sums.
%   Produces valid magic squares for all N > 0 except N = 2.
%   Copyright 1984-2002 The MathWorks, Inc.
%   $Revision: 5.15 $  $Date: 2002/04/15 03:44:23 $
% Historically, MATLAB's magic was a built-in function.
% This M-file uses a new algorithm to generate the same matrices.
n = floor(real(double(n(1))));
% Odd order.
if mod(n,2) == 1
   [J,I] = meshgrid(1:n);
   A = mod(I+J-(n+3)/2,n);
   B = mod(I+2*J-2,n);
   M = n*A + B + 1;
% Doubly even order.
elseif mod(n,4) == 0
   [J,I] = meshgrid(1:n);
   K = fix(mod(I,4)/2) == fix(mod(J,4)/2);
   M = reshape(1:n*n,n,n)';
   M(K) = n*n+1 - M(K);
% Singly even order.
else
   p = n/2;
```

```
    M = magic(p);
    M = [M M+2*p^2; M+3*p^2 M+p^2];
    if n == 2, return, end
    i = (1:p)';
    k = (n-2)/4;
    j = [1:k (n-k+2):n];
    M([i; i+p],j) = M([i+p; i],j);
    i = k+1;
    j = [1 i];
    M([i; i+p],j) = M([i+p; i],j);
end
```

2. save,load 函数存取数据

如果需要把数据存入 MATLAB 的程序中,则需要用 save 命令;取出该数据使用 load 命令。

```
>> sz000001 = struct('date','2007-8-20','close',37.9,'vol',36638584)
sz000001 =
           date: '2007-8-20'
           close: 37.9000
           vol: 36638584
>> save price.mat sz000001                % 将数据保存为文件,名称是 price.mat
>> dir
⋮
price.mat
```

这样数据就被保存为 price.mat 文件。如果要载入该文件的数据到内存中可以使用 load 命令。

```
>> load price
```

再用 who 命令查看内容。

```
>> who
Your variables are:
sz000001
```

3. Help 查询

例如,希望了解 sort 函数的具体用法,可以执行下面操作:

```
>> help sort
```

这样关于 sort 函数的说明就被显示如下:

```
SORT   Sort in ascending or descending order.
    For vectors, SORT(X) sorts the elements of X in ascending order.
    For matrices, SORT(X) sorts each column of X in ascending order.
    For N-D arrays, SORT(X) sorts the along the first non-singleton
    dimension of X. When X is a cell array of strings, SORT(X) sorts
    the strings in ASCII dictionary order.
     ⋮
    See also issorted, sortrows, min, max, mean, median.
    Overloaded functions or methods (ones with the same name in other directories)
       help cell/sort.m
       help sym/sort.m

    Reference page in Help browser
       doc sort
```

4. 浏览网页

在 MATLAB 浏览器中可以浏览相关网页。如果浏览 MathWorks 公司的主页只需在 Command 窗口中输入网址即可。

```
>> web www.mathworks.com.cn
```

MATLAB 浏览器的页面如图 2.6 所示。

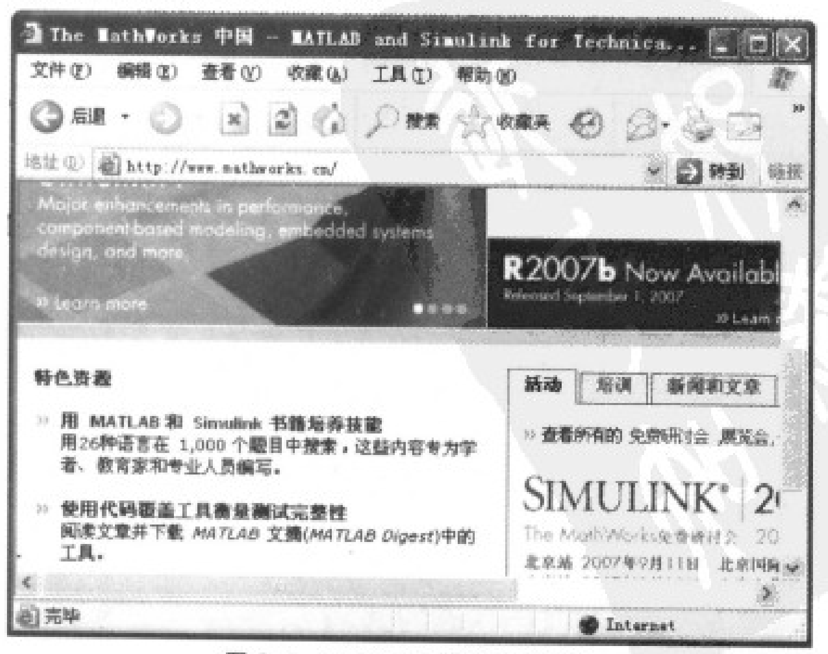


图 2.6　MATLAB 浏览器的页面

5. 文件管理命令

文件管理的命令，包括列出文件名、显示或删除文件、显示或改变当前目录等(what，dir，type，delete，cd，which)。

what：显示当前目录下所有与 MATLAB 相关的文件及它们的路径。

dir：显示当前目录下所有的文件。

which：显示某个文件的路径。

cd path：由当前目录进入 path 目录。

cd .：返回上一级目录。

cd：显示当前目录。

type filename：在命令窗口中显示文件 filename。

delete filename：删除文件 filename。

思考题

1. 对结构数组进行赋值，$s.x$ 内容为矩阵 $\begin{pmatrix}1 & 4\\ 3 & 5\end{pmatrix}$，$s.y$ 为字符串'MATLAB Financal ToolBox'。

2. 利用 diag 函数构建对角矩阵 $\begin{pmatrix}1 & 2 & 3 & 1\\ 0 & 2 & 2 & 3\\ 0 & 0 & 3 & 2\\ 0 & 0 & 0 & 5\end{pmatrix}$。

3. 求解非线性方程组 $\begin{cases}x^3+y=31\\ x+y=7\end{cases}$。

4. 求导数 $y=\dfrac{e^x+x^{1/2}}{x}$。

5. 求矩阵 $\begin{pmatrix}1 & 3.5 & 1.5\\ 0.5 & 3.4 & 6\\ 2.5 & 3 & 2\end{pmatrix}$ 的迹与秩。

6. 求解线性方程组 $\begin{cases}x+y+z=4\\ 2x+y=3\\ y+z=3\end{cases}$。

第3章 固定收益证券的计算

固定收益是非常重要的金融产品，本章重点在于固定收益现金流的计算和利率期限结构的估计。要求读者能运用MATLAB的固定收益工具箱计算现金流现值、将来值、久期与凸度，掌握利用现有的债券品种推出利率期限结构，并且对新国债品种进行定价。

3.1 固定收益证券的基本概念

固定收益证券可以理解为一组稳定现金流的证券，在证券家族中占有重要地位，债券是固定收益证券的主要品种。广义来说，固定收益证券还包括了债券市场上的衍生产品以及优先股。债券是以借贷协议形式发行的证券，一般包含如下几个要素：偿还期、面值和票面利率等。息票债券指发行人在有效期内向持有人支付利息（通常每半年一次）的合约，又称为息票支付。之所以称为息票，是因为在计算机引入金融实务之前，人们通常将债券上的息票剪下邮寄回发行方索取利息。

许多固定收益证券赋予发行者或投资者某些权利，这些权利看上去很简单，但每个权利都需要很多知识诠释，都会使固定收益证券的现金流结构发生变化。如果几个特征集中到一起，则可以使得固定收益证券定价与风险管理变得非常复杂。MATLAB固定收益工具箱用于计算欧美市场的固定收益，我国债券市场和欧美市场存在着一些差别，读者在利用金融工具箱分析固定收益时应注意这一差别。

3.1.1 美国固定收益证券的种类

美国国债是固定收益的重要形式，一般以贴现债券（discount security）与息票债券（coupon bonds）形式发行。贴现债券发行价低于面值，不支付利息，在到期日获取面值金额的收益。息票债券一般按照一定的票息率发行，每隔一段时间支付一次票息，到期按照面值金额赎回。

美国的固定收益证券可分为以下几个品种：

1. 短期国库券

短期国库券T-bills（Treasury bills）是美国政府通过竞标发售的一种短期证券，价格相对面值有折扣，属于贴现发行，其面值一般为1～10万美元，期限为3,6,9个月不等。短期国库券的流动性是最高的，其利率可以作为无风险利率。

2. 政府票据

政府票据 T - note(Treasury note)是美国政府发行的中期国债,期限 1～10 年,以息票方式支付利息。

3. 长期国库券

长期国库券 T - bonds(Treasure bonds)是美国政府发行的长期债券,期限 10 年以上,面值 1～10 万美金,以息票支付利息,通常每半年支付一次利息,在到期日偿还本息。

4. 零息票债券

零息票债券 Z - bonds(Zero - coupon bonds)是买卖价格相对面值有较大折让的企业或市政债券。这类债券在发行时加入折让,或由一家银行除去息票,然后包装成为零息票债券发行,投资者在债券到期以面值赎回时可实现利润。

长期零息券一般是由附带息票票据创造出来的,购买息票国债的经纪人可以要求财政部停止债券的现金支付,使其成为独立证券序列;每一笔证券都具有原始债券收益要求权。例如,一张 10 年期债券被“剥离”成 20 份半年期债券,每张都可视为零息票,它们到期日从 6 个月到 10 年不等,最后本金支付是另一张零息证券,所有支付都单独计算,并配有自己 CUSIP 号码(统一由美国证券鉴定程序委员会颁布)。具有这种标识符的证券都可以在联邦银行及其分支机构上进行网上电子交易,由于这些债券息票被“剥离”了,因此被称为本息剥离式国债 STRIPS(Separate Trading of Registered Interest and Principal of Securities)。从 1987 年 5 月起,美国财政部也允许一系列被剥离债券的票息重新组合成息票。

5. 美国 CD 存单

美国 CD 存单(certificate deposit)是银行等金融机构向存款人发行的证券,存单上标有一个到期日和利率,并且以任意面值发行,可以买卖,属于货币市场工具。偿还期短于 1 年,一般按照“Act/360”规则计算应计天数。CD 存单是按照面值发行,单利计算利率,利息计算公式为

$$利息=100\times c\times D/360$$

式中:c 是票面利率,D 是发行日至到期日之间的天数。

6. 回购协议

回购协议(repurchase agreement)属于短期抵押贷款,是指一方向另一方出售证券的同时,承诺在将来某一天按照商定的价格将相同的证券买回,通常由借款方发起并贷出证券,回购中涉及的证券通常具有较高的信用质量,如国债。回购协议建立了货币市场和债券市场之间的联系。

回购协议的运作机理非常简单，分为两个步骤：第一步，以债券作为抵押借入资金；第二步，经过一段时间，按照约定的价格买回抵押的债券。

7. 可转换债券

可转换债券(convertible security)是一种具有固定收益的证券，其特点是持有者可以将其转换为普通股股票，在合约的条款中规定了可转换债券可以转换为普通股的条件，转换权是否行使取决于持有者的判断力，并且持有者决定何时转换为股票。可转换债券介于普通股和普通债券之间，又称为股票类连接证券。

可转换债券属于次级债券，如果企业破产，则应首先满足优先债权人的要求权，然后满足次级债券的要求权，但是次级债券优于优先股(包括可转换优先股和不可转换优先股)和普通股。总的来讲，可转换债券属权益类证券，其特点是享有先于普通股获得股息偿付的优先权和较高收益，并且有机会分享公司股价上涨时的好处。

可转换债券的价格与很多因素有关，如股票的价格、上市公司信用、股票波动率和利率波动率的变化。

8. 远期利率协议

远期利率协议 FRA(Forward Rate Agreements)是一种远期合约，买卖双方(客户与银行或两个银行同业之间)商定将来某个时间点开始的一定期限的协议利率，并规定以何种利率为参照利率、利息起算日、协议利率、期限和本金。

在这种协议下，交易双方约定从将来某一确定的日期开始在某一特定的时期内借贷一笔利率固定、数额确定，以具体货币表示的名义本金。远期利率协议的买方就是名义借款人，如果市场利率上升，就避免了利率风险；但若市场利率下跌，就会受到损失。远期利率协议的卖方就是名义贷款人，他按照协议确定的利率收取利息；显然，若市场利率下跌，他将受益；若市场利率上升，他则受损。

3.1.2 美国国债报价方式

美国中长期国债、本金分离登记交易(strips)采用分数报价，分数部分用冒号或者连字符“-”分开，例如 95-3/4，表示 95.75 美元。表 3.1 说明百分比报价与其对应的实际价格。

表 3.1 国债报价方式实例

百分比报价	对应小数形式	面值/美元	实际价格/美元
98	0.98	1 000	980
$88_{1/2}$	0.885	10 000	8 850
$951_{1/64}$	0.951 718 8	100 000	95 171.88

续表 3.1

百分比报价	对应小数形式	面值/美元	实际价格/美元
100	1.00	5 000	5 000
$104_{3/4}$	1.047 5	10 000	10 475
$105_{19/32}$	1.055 937	100 000	1 035 937.50
$107_{3/8}$	1.073 75	250 000	257 768.44
106	1.06	1 000	1 060.00

3.1.3 固定收益相关概念

1. 交易日

交易日(trade date)就是买卖双方达成交易的日期。如果通过拍卖方式购入证券,则交易日是拍卖结果被确认、购买者被告知他们分摊数量及价格的日期。如果固定收益证券由一承购集团成员所购买,则交易日与牵头者最终将承销证券分配给成员的日期一致。

2. 结算日

结算日(settle date)一般指买入方支付价格和卖出方交割证券的日期。美国国库券交割日为交易日之后第一个营业日(交割术语为 T+1),交割日也可以由交易者之间商定。如果交割日刚好支付利息,那么债券出售者获得当天利息支付,而债券购买者获得其余款项。有时通过 Fed Wine 机构交割证券,交易日即为交割日。

3. 到期日

到期日(maturity)是指固定收益证券债务合约终止的日期,到期日当日发行人应还清所有本金与利息。很多固定收益证券如定期存款、短期国库券、商业票据、再回购协议、外汇掉期、零息票债券等只有一个到期日,日期计算都以这个到期日为基准。

4. 本　金

本金(principal)有时称为面值(par value),是指固定收益票面金额。

5. 票面利率

票面利率(coupon rate)是指发行人支付给持有人的利息,有时也称名义利率(nominal rate)。票面利率一般指年利率。

6. 月末法则

月末法则(end of month ruler):当债券到期日在某月的最后一天,而且该月天数小于等于30天时有两种情况:一是债券的到期日在每月固定日期支付;二是票息债券在每月的最后一天支付。MATLAB默认的是第二种情况。例如,今天是2005年2月28日,每半年发放一次票息,下一次发放票息可以是2005年8月28日,也可以是2005年8月31日;如果没有月末法则就是2005年8月28日;如果有月末法则是2005年8月31日。

7. 起息日距离交割日天数

起息日到交割日天数DCS(Days from Coupon to Settlement)为从起息日(含)到交割日(不含)之间天数,付息日作为下一个利息期间第一天而不计入DCS。

8. 交割日距离到期日的天数

交割日距离到期日的天数DSM(Days from Settlement to Maturity):一般规则是DSM包括交割日而不包括到期日,这样买方有动力尽早交易,获得当天收益;卖方在交割当天就获得资金使用权,因此到期日当天不能获得利息。

3.1.4 常见应计天数计算方法

1. 应计天数计算方法

应计天数是指债券起息日或上一付息日至结算日的天数,在此期间发生的利息称为应计利息。实务中应计天数有以下几种方法。

- Act/Act:分子分母都是按照实际天数计算,大多数年份是365天,闰年是366天。
- Act/360:按实际天数计息,一年按360天计算。
- Act/365:按实际天数计息,一年按365天计算。
- 30/360(European):每月30天,每年360天,起始日或到期日为31日则改为30日。也可记为30E/360。
- 30/360(ISDA):ISDA为国际互换交易协会(International Swap Dealers Associations)的简称,每月30天,每年360天。起始日或到期日为31日则改为30日。但如果到期日为31日,起始日不为30日、31日,则31日不变。
- 30/360(PSA):PSA为美国公众证券协会(Public Securities Association)的简称,每月30天,每年360天,起始日或到期日为31日则改为30日。但如果到期日为31日,起始日不为30日、31日,则31日不变。起始日为2月最后一天改为30日。
- 30/360(SIA):SIA为美国证券业协会(Securities Industry Association)的简称,应计

天数规则如下。

① 每月 30 天，每年 360 天。如果到期日为 31 日，起始日不为 30 日、31 日，则 31 日不变。

② 如果不是闰年，起始日和到期日都为 2 月 28 日，则二者都改为 30 日；如果是闰年，起始日和到期日都为 2 月 29 日，则二者都改为 30 日。

● Act/365(Japanese)：按实际天数计息，每月 30 天，每年 365 天，不考虑闰年。

30E/360、30/360(ISDA)、30/360(PSA)及 30/360(SIA)之间的区别如表 3.2 所列。

表 3.2 应计天数方法对比表

名　称	内容对比
30E/360	每月 30 天，每年 360 天，起始日或到期日为 31 日则改为 30 日
30/360(ISDA)	同 30E/360。但如果到期日为 31 日，起始日不为 30 日、31 日，则 31 日不变
30/360(PSA)	同 30/360(ISDA)。如果不是闰年，起始日为 2 月 28 日，改为 30 日；如果是闰年，起始日为 2 月 29 日，改为 30 日
30/360(SIA)	同 30/360(ISDA)。如果不是闰年，起始日和到期日都为 2 月 28 日，则二者都改为 30 日；如果是闰年，起始日和到期日都为 2 月 29 日，则二者都改为 30 日

表 3.3 列举了 5 个实例显示各个应计天数法则之间的差别。

表 3.3 应计天数计算方法实例

	30E/360	30/360(ISDA)	30/360(PSA)	30/360(SIA)
2007 年 2 月 27 日－2007 年 2 月 28 日	1	1	1	1
2007 年 2 月 27 日－2007 年 3 月 31 日	33	34	34	34
2007 年 2 月 28 日－2007 年 3 月 1 日	3	3	1	1
2007 年 2 月 28 日－2007 年 3 月 31 日	32	33	30	30
2006 年 2 月 28 日－2007 年 2 月 28 日	360	360	358	360

注：30E/360、30/360(ISDA)与 30/360(SIA)虽然应计天数都为 360 天，但是起止日期不一样。30E/360 与 30/360(ISDA)起止日期为 2006 年 2 月 28 日至 2007 年 2 月 28 日，30/360(SIA)起止日期为 2006 年 2 月 30 日至 2007 年 2 月 30 日。

全球主要债券市场应计天数法则和息票付款频率如表 3.4 所列。

表 3.4　全球主要债券市场应计天数法则和息票付款频率

市　场	息票付款频率	应计天数法则	市　场	息票付款频率	应计天数法则
美国政府债券	每半年	Act/Act	德国政府债券	每年	30E/360
美国公司债券	每半年	30/360	瑞士政府债券	每年	30E/360
美国政府机构债券	每年、每半年、每季	30/360	荷兰政府债券	每年	30E/360
美国市政债券	每半年	30/360	欧洲债券	每年	30E/360
英国政府债券	每半年	Act/365	意大利政府债券	每年	30E/360
澳大利亚政府债券	每半年	Act/Act	法国债券	每年	Act/Act
新西兰政府债券	每半年	Act/Act	丹麦政府债券	每年	30E/360
加拿大政府债券	每半年	Act/Act	瑞典政府债券	每年	30E/360

根据中国人民银行 2007 年 6 月 22 日规定，全国银行间债券市场应计天数基准由“实际天数/365”调整为“实际天数/实际天数”。

需要注意的是应计天数计算时忽略了非营业日（双休日、节假日）因素。例如，2006 年 6 月 25 日发行的长期债券，到了 2006 年 12 月 25 日支付第一次票息，但是由于 12 月 25 日是西方的圣诞节，票息的支付通常在节后第一个营业日完成的。

2. 应计天数

在 MATLAB 中 Act/360，30E/360，30/360（ISDA），30/360（PSA）对应的应计天数函数分别是 daysact，days360e，days360isda，days360psa，计算实际的天数函数是 daysact。

调用方式

```
NumDays = daysact(StartDate, EndDate)
NumDays = days360(StartDate, EndDate)
NumDays = days360e(StartDate, EndDate)
NumDays = days360isda(StartDate, EndDate)
NumDays = days360psa(StartDate, EndDate)
```

输入参数

StartDate	起始日期
EndDate	结束日期

输出参数

NumDays	起始日和终止日之间的间隔天数

下面计算 2000 年 2 月 1 日至 2000 年 3 月 1 日的间隔天数，采用 30E/360 法则。

```
>> StartDate = '1-Feb-2000';
>> EndDate = '1-Mar-2000';
>> NumDays = days360e(StartDate, EndDate)
NumDays =
         30
```

为了方便起见，MATLAB还有统一计算应计期间的函数 daysdif。

调用方式

```
NumDays = daysdif(StartDate, EndDate, Basis)
```

输入参数

StartDate	开始日
EndDate	结束日
Basis	(Optional)应计天数的计算方式。取值内容如下： 0 = Act/Act (默认值) 1 = 30/360(SIA) 2 = Act/360 3 = Act/365 4 = 30/360 (PSA) 5 = 30/360 (ISDA) 6 = 30/360 (European) 7 = Act/365 (Japanese)

输出参数

NumDays	应计天数法则下计算起始日与到期日之间的天数

注意：本书中的其他函数如需输入参数 Basis，如无特别声明，其含义均同本例。

【例3-1】 在Act/Act法则下计算1999年3月1日至2000年3月1日之间的天数。

```
>> NumDays = daysdif('3/1/99', '3/1/00', 1)
   NumDays =
          360
```

进一步地，起始日为1998年3月1日，结算日为2001年3月1日、2002年3月1日、2003年3月1日，分别计算应计天数。

```
>> MoreDays = ['3/1/2001'; '3/1/2002'; '3/1/2003'];
>> NumDays = daysdif('3/1/98', MoreDays)
   NumDays =
        1096
```

```
1461
1826
```

3. 计算债券距离到期日还需要支付的票息次数

在实务中有时需要计算债券距离到期日期间支付票息的次数，在 MATLAB 中执行此项功能的函数是 cpncount。

调用方式

```
NumCouponsRemaining = cpncount(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate, First-
                      CouponDate, LastCouponDate, StartDate)
```

输入参数

Settle　债券的结算日

Maturity　债券的到期日

Period　(Optional)每年支付票息次数，Optional 表示可选项，表示该项内容没有或者和默认值相同，可以不填。可以选择的值为 0,1,2,3,4,6,12，其含义如下：

0	不支付票息，到期日支付
1	每年支付 1 次票息
2	(默认值)每半年支付 1 次票息
3	每 4 个月支付 1 次票息
4	每季度支付 1 次票息
6	每 2 个月支付 1 次票息
12	每月支付 1 次票息

Basis　(Optional)应计天数的计算法则

0 = Act/Act (默认)
1 = 30/360(SIA)
2 = Act/360
3 = Act/365
4 = 30/360 (PSA)
5 = 30/360 (ISDA)
6 = 30/360 (European)
7 = Act/365 (Japanese)

EndMonthRule　(Optional)月末法则。仅对到期日是 30 日或者小于 30 日有效，0 表示发放票息的日期相同；1(默认值)表示票息在月的最后一天发放

IssueDate　(Optional)　发行日

FirstCouponDate　(Optional)　首次支付票息日期

LastCouponDate　(Optional)　最后一次支付票息日期

StartDate (Optional) 起始日

输出参数

NumCouponsRemaining 剩下的票息支付次数

注意:(Optional)表示可选项,表示该项内容没有或者和默认值相同。

【例 3-2】 债券的起始日是1997年9月14日,到期日分别是2000年9月30日、2001年10月31日以及2002年11月30日,问该债券将来支付票息的次数。

```
>> Maturity = ['30 Sep 2000'; '31 Oct 2001'; '30 Nov 2002'];
>> NumCouponsRemaining = cpncount('14 Sep 1997', Maturity)
NumCouponsRemaining =
                     7
                     9
                     11
```

4. 计算前一个票息支付日

调用方式

```
PreviousCouponDate = cpndatep(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate, First-
                     CouponDate, LastCouponDate)
```

输入参数

Settle	结算日
Maturity	到期日
Period	(Optional)每年支付票息次数,可以选择的值为0,1,2(默认值),3,4,6,12
Basis	(Optional)应计天数法则
EndMonthRule	(Optional)月末法则
IssueDate	(Optional)发行日
FirstCouponDate	(Optional)初次付息日
LastCouponDate	(Optional)最后一次付息日

输出参数

PreviousCouponDate 前一个支付票息的日期

下面是一个例子,债券的结算日是1997年3月14日,到期日是2000年11月30日,每半年支付1次票息,不采用月末法则,计算前一个票息日。

```
>> Settle = '14 Mar 1997';
>> Maturity = '30 Nov 2000';
```

```
>> PeriodPeriod = 2;
>> Basis = 0;
>> EndMonthRule = 0;
>> NextCouponDate = cpndatep(Settle,Maturity, PeriodPeriod, EndMonthRule);
>> datestr(NextCouponDate)
ans =
     30 - Nov - 1996
```

5. 计算下一个支付票息的日期

在 MATLAB 中计算下一个支付票息的日期函数是 cpndaten。

调用方式

```
NextCouponDate = cpndaten(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate, FirstCou-
                 ponDate, LastCouponDate)
```

输入参数

Settle	结算日
Maturity	到期日
Period	(Optional)每年支付票息次数,可以选择的值为 0,1,2(默认值),3,4,6,12
Basis	(Optional)应计天数法则
EndMonthRule	(Optional)月末法则
IssueDate	(Optional)发行日
FirstCouponDate	(Optional)初次付息日
LastCouponDate	(Optional)最后一次付息日

输出参数

NextCouponDate	下一个支付票息的日期

下面是一个例子。

```
>> NextCouponDate = cpndaten('14 Mar 1997', '30 Nov 2000', 2, 0, 0)
NextCouponDate =
           729540
>> datestr(NextCouponDate)
ans =
     30 - May - 1997
```

【例 3-3】 结算日是 1997 年 3 月 14 日,到期日分别是 2000 年 9 月 30 日、2000 年 10 月 31 日以及 2000 年 11 月 30 日,计算下一个付息日。

```
>> Maturity = ['30 Sep 2000'; '31 Oct 2000'; '30 Nov 2000'];
>> NextCouponDate = cpndaten('14 Mar 1997', Maturity);
>> datestr(NextCouponDate)
ans =
      31 - Mar - 1997
      30 - Apr - 1997
      31 - May - 1997
```

6. 计算下一个似票息的日期

有时票息支付并不规则，例如，第一次付息日和最后一次付息日与票息发放日不规则。可以用 cpndatenq 函数计算下一个似票息日天数。

调用方式

```
NextQuasiCouponDate = cpndatenq(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate,
                      FirstCouponDate, LastCouponDate)
```

输入参数同前

输出参数

NextQuasiCouponDate　　下一个似票息的发放日期

对于规则的债券 cpndaten 和 cpndatenq 计算的结果相同。例如，两个债券，结算日分别为 1997 年 5 月 30 日、1997 年 12 月 10 日，到期日分别为 2002 年 11 月 30 日、2004 年 6 月 10 日。计算下一个似票息日如下：

```
>> Settle = char('30 - May - 1997','10 - Dec - 1997');
>> Maturity = char('30 - Nov - 2002','10 - Jun - 2004');
```

先用 cpndaten 函数计算下一个票息日。

```
>> NextCouponDate = cpndaten(Settle, Maturity);
>> datestr(NextCouponDate)
ans =
      31 - May - 1997
      10 - Jun - 1998
```

用 cpndatenq 函数计算下一个票息日。

```
>> NextQuasiCouponDate = cpndatenq(Settle, Maturity);
>> datestr(NextQuasiCouponDate)
ans =
```

```
      31 - May - 1997
      10 - Jun - 1998
```

可以看出二者计算结果相同。下面增加第一次付息日，然后计算下一次付息日。

```
>> FirstCouponDate = char('30 - Nov - 1997','10 - Dec - 1998');
>> NextQuasiCouponDate = cpndatenq(Settle, Maturity, 2, 0, 1, [],FirstCouponDate);
>> datestr(NextQuasiCouponDate)
ans =
      30 - Nov - 1997
      10 - Dec - 1998
```

可见如果改变第一次付息日，计算的结果就不一样。

7. 计算票息日期

在 MATLAB 中计算票息日期的函数是 cfdates。

调用方式

```
CFlowDates = cfdates(Settle, Maturity, Period, Basis, EndMonthRule,
IssueDate, FirstCouponDate, LastCouponDate, StartDate)
```

输入参数

Settle	结算日
Maturity	到期日
Period	(Optional)付息频率
Basis	(Optional)应计天数法则
EndMonthRule	(Optional)月末法则
IssueDate	(Optional)发行日
FirstCouponDate	(Optional)首次付息日
LastCouponDate	(Optional)最后一次付息日
StartDate	(Optional) 现金流开始日，默认值为结算日

输出参数

CFlowDates	票息日期

【例 3 - 4】 结算日是 1997 年 3 月 14 日，到期日是 1998 年 11 月 30 日，计算现金流日期。

```
>> CFlowDates = cfdates('14 Mar 1997', '30 Nov 1998', 2, 0, 1)
CFlowDates =
          729541       729724       729906       730089
>> datestr(CFlowDates)
```

```
ans =
      31 - May - 1997
      30 - Nov - 1997
      31 - May - 1998
      30 - Nov - 1998
```

下面计算起始日相同而到期日不同债券的现金流发生日。

```
>> Maturity = ['30 - Sep - 1997'; '31 - Oct - 1998'; '30 - Nov - 1998'];
>> CFlowDates = cfdates('14 - Mar - 1997', Maturity)
CFlowDates =
         729480      729663      NaN      NaN
         729510      729694      729875   730059
         729541      729724      729906   730089
```

下面显示上述3种债券现金流具体日期。

```
>> datestr(CFlowDates(1,1:2))
ans =
      31 - Mar - 1997
      30 - Sep - 1997
>> datestr(CFlowDates(2,:))
ans =
      30 - Apr - 1997
      31 - Oct - 1997
      30 - Apr - 1998
      31 - Oct - 1998
>> datestr(CFlowDates(3,:))
ans =
      31 - May - 1997
      30 - Nov - 1997
      31 - May - 1998
      30 - Nov - 1998
```

3.1.5 全价与净价

1. 全价和净价

债券报价方式分全价与净价两种。全价(full price,dirty price)包括了应计利息(accrued interest),净价(clean price)等于全价减去应计利息,即净价=全价-应计利息。

应计利息计算公式如下：

$$AI=c\times\frac{\text{上次付息日到交割日之间的天数}}{\text{应计期间}}$$

式中：AI 为应计利息，c 为每半年支付金额。上次付息日到交割日之间的天数称为应计期间。

虽然债券报价分全价与净价两种方式，但是买方支付的必须是全价。下面以每年支付两次票息的债券为例给出全价计算公式。

$$tp=\frac{c}{(1+y)^{W}}+\frac{c}{(1+y)^{W+1}}+\cdots+\frac{c}{(1+y)^{n-1+W}}+\frac{M}{(1+y)^{n-1+W}}$$

式中：tp 是全价，c 为半年度票息，y 是到期收益率/2，M 是到期日现金流，n 为剩余的票息付款次数，$W=\frac{\text{结算日至下一个票息付款日之间的天数}}{\text{票息期间的天数}}$。

【例 3-5】 公司债券发行日是 2000 年 3 月 1 日，到期日 2006 年 3 月 1 日，每年支付两次利息，结算日是 2000 年 7 月 17 日，票息率为 10 %，面值 100 元，结算日和下一个付息日(2000 年 9 月 1 日)之间天数为 44 天，计息周期天数为 180 天，从上一个付息日(2000 年 3 月 1 日)到交割日之间天数为 136(180－44)天(天数计算规则是 30E/360)。应计利息为

$$AI=5\times\frac{136}{180}=3.777\ 8$$

在美国，债券交易按净价报价，其他市场一般按全价报价。

2. 应计天数因子

在 MATLAB 中计算应计天数因子的函数是 accrfrac，计算公式如下：

$$\text{accrfrac}=\frac{\text{上次付息日到交割日之间的天数}}{\text{计算周期天数}}$$

调用方式

```
Fraction = accrfrac(Settle, Maturity)
Fraction = accrfrac(Settle, Maturity, Period)
Fraction = accrfrac(Settle, Maturity, Period, Basis)
Fraction = accrfrac(Settle, Maturity, Period, Basis, EndMonthRule)
Fraction = accrfrac(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate)
Fraction = accrfrac(Settle, Maturity, Period, Basis, EndMonthRule,IssueDate,
           FirstCouponDate)
Fraction = accrfrac(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate,
           FirstCouponDate, LastCouponDate)
Fraction = accrfrac(Settle, Maturity, Period, Basis, EndMonthRule,IssueDate,
           FirstCouponDate, LastCouponDate, StartDate)
```

输入参数

Settle	结算日
Maturity	到期日
Period	(Optional) 付息频率
Basis	(Optional) 应计天数法则
EndMonthRule	(Optional) 月末法则
IssueDate	(Optional) 发行日
FirstCouponDate	(Optional) 第一个付息日
LastCouponDate	(Optional) 最后一个付息日
StartDate	(Optional) 起始日

输出参数

Fraction	应计天数因子

【例 3-6】 有 3 种固定收益，结算日为 1997 年 3 月 14 日，到期日分别为 2000 年 11 月 30 日、2000 年 12 月 31 日和 2001 年 1 月 31 日。试计算应计天数因子。

```
>> Settle = '14-Mar-1997';
>> Maturity = ['30-Nov-2000'
              '31-Dec-2000'
              '31-Jan-2001'];
>> Period = 2;
>> Basis = 0;
>> EndMonthRule = 1;
>> Fraction= accrfrac(Settle, Maturity, Period, Basis, EndMonthRule)
Fraction =
         0.5714
         0.4033
         0.2320
```

然后根据下面公式可以算得应计利息。

$$应计利息=Fraction\times 面值\times(票息率/付息频率)$$

下面验算**【例 3-5】**。

```
>> Settle='17-jul-2000';
>> Maturity='1-sep-2006';
>> CouponRate=0.1;
>> Period=2;
>> Basis=6;
```

```
>> EndMonthRule = 0;
>> Fraction = accrfrac(Settle, Maturity, Period, Basis, EndMonthRule)
Fraction =
          0.7556
```

应计利息＝Fraction×面值×(票息率/付息频率)＝0.755 6×100×(0.1/2)＝3.777 8

3. 应计利息

在MATLAB中计算应计利息的函数是acrubond。

调用方式

```
AccruInterest = acrubond(IssueDate, Settle, FirstCouponDate, Face, CouponRate, Period, Basis)
```

输入参数

IssueDate	发行日
Settle	结算日
FirstCouponDate	首次付息日
Face	面值
CouponRate	票息率
Period	(Optional) 付息频率
Basis	(Optional) 应计天数法则

输出参数

AccruInterest	应计利息

下面验算【例3-5】的结果。

```
>> IssueDate = '1 - Mar - 2000';
>> Settle = '17 - jul - 2000';
>> FirstCouponDate = '1 - sep - 2000';
>> Face = 100;
>> CouponRate = 0.1;
>> Period = 2;
>> Basis = 6;
>> AccruInterest = acrubond(IssueDate, Settle, FirstCouponDate, Face, CouponRate, Period, Ba-
                  sis)acrubond('1 - Mar - 2000','17 - jul - 2000','1 - sep - 2000',100,0.1,2,6)
AccruInterest =
          3.7778
```

3.1.6 贴现率计算

MATLAB 根据票息支付的频率计算不同情况下的贴现率。贴现率公式如下：

$$\text{disc}=\frac{1}{\left(1+\frac{R}{F}\right)^{T}}$$

式中：R 为票息率，T 为时间，disc 为贴现率。F 为一年中计息频率，其内容如下：

1	每年支付 1 次票息。
2	每半年支付一次票息。
3	每 4 个月支付一次票息。
4	每季度支付一次票息。
6	每 2 个月支付一次票息。
12	每月支付一次票息。
365	每天支付一次票息。

对于复合利率，MATLAB 把 $F=-1$ 当成是复合利率，其贴现率计算公式如下：

$$\text{disc}=e^{-T\cdot Z}$$

式中：T 为时间，Z 为复合利率。

3.1.7 时间因子与付息次数

1. 根据结算日和到期日计算时间因子

调用方式

```
[TFactors, F] = date2time(Settle, Maturity, Compounding, Basis, EndMonthRule)
```

输入参数

Settle 结算日

Maturity 到期日

Compounding 一年中支付利息的频率。可以选 1,2,3,4,6,12

disc 计算方式如下：

$$\text{disc}=\frac{1}{\left(1+\frac{R}{F}\right)^{T}}$$

式中：F 为一年中计息频率，R 为票息率，T 为时间因子，disc 为贴现率

Basis (Optional) 应计天数法则

EndMonthRule (Optional) 月末法则

输出参数

TFactors　　　时间因子 T

F　　　每年付息次数

【例 3-7】 3 种债券的结算日为 1997 年 3 月 14 日，到期日分别为 2000 年 11 月 30 日、2000 年 12 月 31 日和 2001 年 1 月 31 日，每年支付 2 次利息，应计天数法则为 Act/Act，采用月末法则，试计算时间因子。

```
>> Settle = '14-Mar-1997';
>> Maturity = ['30-Nov-2000';'31-Dec-2000';'31-Jan-2001'];
>> Compounding = 2
>> Basis = 0;
>> EndMonthRule = 1;
>> [TFactors, F] = date2time(Settle, Maturity, Compounding, Basis, EndMonthRule)
TFactors =
          7.4286
          7.5967
          7.7680
F =
    2
```

2. 根据结算日和时间因子推算到期日

调用方式

```
Dates = time2date(Settle, TFactors, Compounding, Basis, EndMonthRule)
```

输入参数、输出参数同前。

下面根据上面的例子推算出到期日。

```
>> Dates_calc = time2date(Settle, TFactors, F, Basis, EndMonthRule);
>> datestr(Dates_calc)
ans =
      30-Nov-2000
      31-Dec-2000
      31-Jan-2001
```

3. 由利率转换成相应的贴现率

调用方式

```
disc = rate2disc(Compounding, Rates, EndTimes)
```

输入参数

Compounding	一年中支付利息的频率。可以选 1,2,3,4,6,12 disc 计算方式如下： $$\text{disc} = \frac{1}{\left(1+\frac{R}{F}\right)^{T}}$$ 式中：F 为一年中付息频率，R 为利率，T 为时间(用期间数表示)，disc 为贴现率
Rates	年利率，用小数表示，如 5 % 写成 0.05
EndTimes	结束时间，注意 EndTimes 的单位与 Compounding 一致。例如，同样是 2 年，Compounding = 2 时，表示半年支付利息一次，2 年等于 4 个半年，EndTimes 就必须取 4；Compounding = 4 时，表示一季度支付利息一次，2 年等于 8 个半年，EndTimes 就必须取 8

输出参数

disc	贴现率

【例 3-8】 一种债券年利率为 0.05，半年支付一次利息，到期日为 2 年，记其贴现率为 disc，在 MATLAB 中执行以下命令：

```
>> R = 0.05;F = 2;T = 4;
>> disc = rate2disc(F, R, 4)
disc =
        0.9060
```

计算出贴现率为 0.9060。

调用方式

```
disc = rate2disc(Compounding, Rates, EndDates, StartDates,ValuationDate)
```

输入参数

Compounding	一年中支付利息的频率
Rates	年利率，以小数表示
StartDates	债券开始计息日
EndDates	债券到期日，输入日期型字符
ValuationDate	评估日，默认值等于 StartDates

【例 3-9】 如果一种债券，年利率为 5 %，发行日为 15 - Feb - 2002 年，到期日为 15 - Aug - 2002 年，半年支付一次利息，需要计算 15 - Feb - 2002 到 15 - Aug - 2002 的贴现率，则其贴现率计算如下：

```
>> Compounding = 2;
>> Rates = 0.05;
>> EndDates = '15 - Aug - 2002';   StartDates = '15 - Feb - 2002'; ValuationDate = '15 - Feb - 2002';
>> Disc  = rate2disc(Compounding, Rates, EndDates, StartDates, ValuationDate)
Disc =
       0.9756
```

贴现率为 0.975 6。如果需要计算 2002 年 4 月 15 日到 2002 年 8 月 15 日的贴现率，则只需改动 ValuationDate 即可。

【例 3-10】 已知 3 种债券每月支付一次票息，票息率分别为 0.05，0.04，0.06，起始日与结束日都是按照支付票息时间段计算。第一种债券从第 0 期到第 12 期；第二种债券从第 6 期到第 18 期；第三种债券从第 12 期到第 24 期，计算债券的贴现率。

```
>> Compounding  =  12;
>> Rates  =  [0.05; 0.04; 0.06];
>> EndTimes  =  [12; 18; 24];
>> StartTimes  =  [0; 6; 12];
>> Disc  =  rate2disc(Compounding, Rates, EndTimes, StartTimes)
Disc =
       0.9513
       0.9609
       0.9419
```

4. 将票息率折算为每月付息的年回报率

假设年利率为 Rate，每年有 NumPeriods 时间段，每月付息的年回报率为 Return，则有公式

$$\text{Return}=\left(1+\frac{\text{Rate}}{\text{NumPeriods}}\right)^{\text{NumPeriods}}-1$$

在 MATLAB 中将年回报率转换为每月付息的年收益率函数是 effrr 函数。

调用方式

```
Return  =  effrr(Rate, NumPeriods)
```

输入参数

Rate	债券的年回报率
NumPeriods	年支付利率的次数

输出参数

Return 每月付息的年收益率

【例 3-11】 一项投资为 9 年,年回报为 9 %,问平均每月付息的年回报率是多少?

首先把 9 %写成小数形式,然后调用 effrr 函数,在 MATLAB 中执行命令如下:

```
>> Return = effrr(0.09, 12)
Return =
        0.0938
```

这样每月付息的年回报率为 9.38 %。

3.1.8 绝对利差、静态利差和期权调整后的利差

不同债券的到期收益率之差称为收益率溢价,或者利差(yield spread)。绝对利差指的是两个债券品种到期收益率之差,例如某个债券到期收益率为 5%,同期国债收益率为 4 %,那么绝对利差是 1 %(5 %-4 %)。

静态利差(static spread)是指假定投资者持有债券到了偿还期,债券所实现的收益会比同期国债高多少。静态利差不是公司债券到期收益率与国债收益率简单相减,而是衡量债券到期收益率曲线超过国债到期收益率曲线程度。静态利差也被称为 Z-利差,Z 取 Zero 的第一个字母,是指波动率为零时利差。静态利差需要求解下列方程:

$$P_0 = \sum_{t=1}^{N} \frac{C_t}{(1+r_t+r_{ss})^t} + \frac{F}{(1+r_N+r_{ss})N}$$

式中:r_{ss}为静态利差。

由于利差没有考虑到某些债券的含权属性,因此该指标存在一定问题。例如,一个公司债券到期收益率为 8 %,而同期国债收益率为 6 %,这里存在两个百分点利差,但是如果该公司是含权的(债券上面附加了一些保证条款),比如可以提前回购,这时利差就属于期权价值,称为期权调整利差 OAS(Option Adjusted Spread)。

3.2 固定收益函数的调用方法

3.2.1 SIA 基本框架

MATLAB 固定收益工具箱的部分函数是根据美国证券业协会(SIA)的规则制定的,固定收益工具箱中大多数函数输入参数包含下面的内容,具体如表 3.5 所列。

表 3.5　固定收益函数输入参数含义

输入参数	含　义
债券的基本特征	
Settle	结算日
Maturity	到期日
Period	票息支付期间
Basis	应计天数法则
EndMonthRule	月末法则
IssuDate	债券发行日
StartDate	起始日
FirstCouponDate	首次票息日
LastCoupondDate	最后一次付息日
CouponRate	票息率
Face	面值
现金流的特征	
CFlowAmounts	各期现金流的数量
CFlowDates	现金流发生的日期
TFactors	时间因子
CFlowFlags	现金流的特征
债券的数量特征	
Accrfrac	应计天数因子
日期参数	
NumCouponsRemaining	距离到期日剩下的票息次数
NextCouponDate	下一个票息支付日
PreviousCoupondate	上一个或当前票息支付日
NumDaysPeriod	相邻两次付息日之间的天数
NumDaysNext	距离下一个票息支付日的天数
NumDaysPrevious	上一个付息日以来的天数

表 3.5 中的输入参数，只有 Settle 和 Maturity 是必须的，其他都是可选的(Optional)，如果不输入时采用默认值。例如不输入 FirstCouponDate 与 LastCouponDate 时，首次付息日和最后一次付息日按常规方式处理。

MATLAB 在对日期进行计算时首先检查日期的合理性，例如结算日不能晚于到期日。

在 SIA 协会规定的框架下所有固定收益品种包括零息券的收益率、价格换算成与收益率一样都是以半年支付一次票息的方式标价，例如 bndprice，bndyield，cfamount 函数。在计算久期和凸度时都是假设债券每年付息两次。SIA 计算规则还适用下面的情况：

- 计算现金流的日期、现金流数量和现金流日期转化；
- 到期价格以及收益率；
- 久期与凸度。

3.2.2 SIA 框架下默认参数用法

cfdate 函数是计算固定收益函数的现金流支付日期。下面以 cfdate 函数为例，说明如何处理默认值。其完整的调用方式如下：

```
CFlowDates = cfdates(Settle, Maturity, Period, Basis, EndMonthRule, IssueDate, FirstCouponDate,
            LastCouponDate)
```

如果采用下面的方式调用：

```
CFlowDates = cfdates(Settle, Maturity)
```

其他没有输入的参数都视为默认值，Basis＝0，Period＝2，EndMonthRule＝1，IssuDate，FirstCouponDate，LastCoupondDate 为常规债券方式。也可采用下面的调用方式：

```
Settle = '20 - Sep - 1999'
Maturity = '15 - Oct - 2007'
Period = 2
Basis = 0
EndMonthRule = 1
IssueDate = NaN
FirstCouponDate = NaN
LastCouponDate = NaN
```

其中 IssuDate，FirstCouponDate，LastCoupondDate 的值为 NaN（也可以用[ ]代替），在处理时会忽略其内容，按照常规的方法处理。下面的调用方式都是相同的。

```
cfdates(Settle, Maturity)
cfdates(Settle, Maturity, Period)
cfdates(Settle, Maturity, Period, [ ])
cfdates(Settle, Maturity, [ ], Basis)
cfdates(Settle, Maturity, [ ], [ ])
cfdates(Settle, Maturity, Period, [ ], EndMonthRule)
cfdates(Settle, Maturity, Period, [ ], NaN)
```

```
cfdates(Settle, Maturity, Period, [ ], [ ], IssueDate)
cfdates(Settle, Maturity, Period, [ ], [ ], IssueDate, [ ], [ ])
cfdates(Settle, Maturity, Period, [ ], [ ], [ ], [ ],LastCouponDate)
cfdates(Settle, Maturity, Period, Basis, EndMonthRule,IssueDate,...
FirstCouponDate, LastCouponDate)
```

3.2.3 多个债券的调用规则

对于多种债券的输入参数，相同的部分可以共用一个值。不相同时用一组向量分别标明。例如有两种债券，结算日相同，到期日不同，其内容如下：

```
Settle = '20 - Sep - 1999'
Maturity = ['15 - Oct - 2007'; '15 - Oct - 2010']
```

因为结算日相同，可以用一个值，到期日各不相同就用一个向量标明。可以采用下面的方式调用 cfdates 函数分别计算其现金流发生日期。

```
cfdates(Settle, Maturity, 2)
cfdates(Settle, Maturity, [2 2])
cfdates(Settle, Maturity, [])
cfdates(Settle, Maturity, NaN)
cfdates(Settle, Maturity, [NaN NaN])
cfdates(Settle, Maturity)
```

前面两种调用方式表明 Period=2，如果输入一个 2，则默认为所有债券的 Period 都设定为 2。上面的调用方式也是等价的。

3.3 现金流计算

3.3.1 基本概念

1. 现值与终值

现值(present value)是指将来的 1 元钱相当于现在多少钱；终值(future value)是指现在的 1 元钱，将来值多少钱。

2. 即期利率与远期利率

即期利率(spot interest rate)是指某个给定时点上无息债券到期收益率。即期利率可以

看作是一个即期合约的利率水平，这种合约一旦签订，资金立即从债权人转移到借款人手里，由借款人在将来某个特定时点按照合约中标明的利率水平连本带利全部还清。

如果投资者以价格 P_1 购买期限为 n 年无息债券，债券到期后从发行人那里获得一次性现金支付为 M_n，那么第 n 年期即期利率 r_n 与 P_1、M_n 之间关系如下：

$$P_1 = \frac{M_n}{(1+r_n)^n}$$

对于期限较长附息债券，即期利率确定方式有所不同。例如，某投资者以 P_2 价格购买期限为两年、面值为 F 附息债券，每年支付利息为 C，在这种情况下，通常用一年期无息债券来计算一年期即期利率 r_1，那么两年期即期利率 r_2 计算需要求解下面的方程：

$$P_2 = \frac{C}{(1+r_1)^1} + \frac{C+F}{(1+r_2)^2}$$

远期利率(forward interest rate)是指未来两个时点之间利率水平。远期利率可以看作是与一个远期合约有关的利率水平。一个远期利率在现在签订合约中规定，但与未来一段时期有关，这也就是说，远期合约利率条件现在已经确定，只是实际交割将在以后进行。

3. 零息利率

零息利率(zero - rate)是零息债券持有到期收益率，如果市场上有各种不同期限零息债券，很容易衡量零息利率。然而，现实中零息债券非常少，一般通过货币市场和资本市场中其他工具估算零息利率。例如，通过平价收益率曲线建立零息利率曲线，这个过程称为“剥离”方法。

4. 当前收益率

当前收益率(current yield)为债券每年支付利息和价格之比。例如，债券的价格为 100，每年支付利息为 5 元，则当前收益率为 5/100=5 %。

5. 到期收益率

固定收益证券对应于一系列现金流组合，投资者在初始时刻支付一笔现金，将来时刻可以收到利息，如果使用某个收益率对将来现金流进行贴现，而贴现结果正好等于初始时刻支付的现金，这个收益率就是到期收益率 YTM(Yield To Maturity)。到期收益率隐含两个假设：一是投资者持有到期；二是利息再投资的利率是不变的。到期收益率又称内部收益率。

3.3.2 现金流基本计算

1. 现金流现值

现金流现值计算公式如下：

$$PV = P_0 + \frac{P_1}{(1+r)} + \frac{P_2}{(1+r)^2} + \frac{P_3}{(1+r)^3} + \cdots + \frac{P_n}{(1+r)^n}$$

式中：P_i 为第 i 期现金流序列，r 为贴现率。现值计算过程也称为折现，因此现值有时称为折现值(discounted value)，利率 r 称为折现率(discount rate)。

调用方式

```
PresentVal = pvvar(CashFlow, Rate)
PresentVal = pvvar(CashFlow, Rate, IrrCFDates)
```

输入参数

CashFlow	各期的现金流序列
Rate	贴现率
IrrCFDates	现金流发生的日期

输出参数

PresentVal	现金流的现值

【例 3-12】 一项投资各年现金流如表 3.6 所列，贴现率为 0.08，求其现值。

表 3.6　各年现金流

年　份	当　前	第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
金额/元	−10 000	2 000	1 500	3 000	3 800	5 000

记各期现金流为 CashFlow，把现金流 CashFlow 写成向量形式[−10000 2000 1500 3000 3800 5000]，执行下面命令：

```
>> CashFlow = [ - 10000 2000 1500 3000 3800 5000];Rate = 0.08;
>> pvvar(CashFlow, Rate)
ans =
      1.7154e + 003
```

计算现值为 1 715.4。

注意 pavvar 函数计算现值时必须输入当前时刻现金投入。例如，在【例 3-12】中如果直接计算第 1 年、第 2 年、第 3 年、第 4 年及第 5 年现值，则必须将当前时刻现金流设为 0，在 MATLAB 中执行命令如下：

```
>> CashFlow = [0 2000 1500 3000 3800 5000];
>> pvvar(CashFlow, 0.08)
ans =
      1.1715e + 004
```

【例 3-13】 表 3.7 为一项投资现金流及其对应日期，贴现率为 0.09，求其现值。

表 3.7　各个日期发生的现金流

现金发放日	12-Jan-1987	14-Feb-1988	3-Mar-1988	14-Jun-1988	1-Dec-1988
金额/元	-10 000	2 500	2 000	3 000	4 000

与【例 3-12】不同的是，这里给定了现金流发生日期，把日期存入变量 IrrCFDates，在 Command 窗口中执行如下命令：

```
>> CashFlow = [-10000, 2500, 2000, 3000, 4000];
>> Rate = 0.09;
>> IrrCFDates = ['01/12/1987'
                 '02/14/1988'
                 '03/03/1988'
                 '06/14/1988'
                 '12/01/1988'];
>> PresentVal = pvvar(CashFlow, Rate, IrrCFDates)
PresentVal =
            142.1648
```

计算出现值为 142.164 8。

如果将 IrrCFDates 转换成序数日期，结果也一样。

```
>> dnum = datenum(IrrCFDates)     % 将 IrrCFDates 转换成序数日期
dnum =
        725749
        726147
        726165
        726268
        726438
>> PresentVal = pvvar(CashFlow, Rate, dnum)
PresentVal =
            142.1648
```

2. 计算现金流终值

现值考虑的时点是现在，而终值考虑的时点是债券到期日。在 MATLAB 中用 fvvar 函数计算终值。

调用方式

```
FutureVal = fvvar(CashFlow, Rate, IrrCFDates)
```

输入参数、输出参数同 pvvar 函数。

【例 3-14】 现金流同**【例 3-12】**,求该现金流终值。可以在 MATLAB 中执行如下命令:

```
>> CashFlow = [-10000 2000 1500 3000 3800 5000];Rate = 0.08;
>> FutureVal = fvvar(CashFlow, Rate)
FutureVal =
           2.5205e+003
```

如果将 FutureVal 贴现到当前时刻就是现值,代码如下:

```
>> FutureVal/(1+Rate)^5
ans =
      1.7154e+003
```

贴现结果同**【例 3-12】**。

3. 计算年金利率

如果每次的现金流都是一样的,这样的现金流称为年金;如果现金支付发生在年初称为及时年金,发生在年末称为普通年金。

调用方式

```
Rate = annurate(NumPeriods, Payment, PresentValue, FutureValue, Due)
```

输入参数

NumPeriods	偿还期间数
Payment	每次偿还的金额
PresentValue	年金的现值
FutureValue	(Optional)年金终值,默认值是 0
Due	(Optional)普通年金 Due = 0(默认值),及时年金 Due = 1

输出参数

Rate	年金的收益率

【例 3-15】 有一笔贷款金额 5 000 元,4 年还清,每月 130 元,求月利率。4 年合计 4×12=48 个月。

```
>> Rate = annurate(4*12, 130, 5000, 0, 0)
```

```
Rate =
         0.0094
```

该笔贷款月利率为 0.94 %，年贷款利率为

```
>> 12 * Rate
ans =
       0.1132
```

即年利率为 11.32 %。

4. 计算年金期间数

调用方式

```
NumPeriods = annuterm(Rate, Payment, PresentValue, FutureValue, Due)
```

输入参数、输出参数同上。

【例 3-16】 如果一个储蓄账户中已经有了 1 500 美元，计划每月存入 200 美元，利率为 4 %，问多少月后可以达到 6 000 美元？

```
>> NumPeriods = annuterm(0.04/12, 200, 1500, 6000, 0)
NumPeriods =
              21.2206
```

大概 21.220 6 个月后可以达到 6 000 美元。

5. 计算内部收益率

内部收益率是使投资现金流现值等于价格的收益率，其计算公式为

$$P=\frac{C_1}{(1+r)}+\frac{C_2}{(1+r)^2}+\frac{C_3}{(1+r)^3}+\cdots+\frac{C_N}{(1+r)^N}$$

式中：C_i 为第 i 年度现金流，P 为债券价格，N 为年数，r 为内部收益率。

在 MATLAB 中计算内部收益率的是 irr 函数。

调用方式

```
Return = irr(CashFlow)
```

输入参数

CashFlow	从第 0 期开始各期现金流

输出参数

Return	内部收益率

【例 3-17】　一项投资各期现金流如表 3.8 所列。

表 3.8　各期投资现金流

期　限	第 0 期	第 1 期	第 2 期	第 3 期	第 4 期
金额/元	-5 000	1 000	2 000	3 000	4 000

问该项投资内部收益率是多少？

把各期现金流保存在变量 CashFlow 中，把 CashFlow 写成向量形式[-5000　1000　2000　3000　4000]，计算该项目内部回报率，执行如下命令：

```
>> CashFlow = [-5000 1000 2000 3000 4000]
CashFlow =
        -5000        1000        2000        3000        4000
>> irr(CashFlow)
ans =
    0.2727
```

则该项投资的内部收益率为 27.27 %。

【例 3-18】　某公司债券票面利率为 8 %，1 年支付一次利息，期限为 5 年，价格为 103.7 元，同期的国债收益率期限结构如表 3.9 所列。

表 3.9　国债利率期限结构

年	1	2	3	4	5
到期收益率	4.51	4.68	4.84	4.99	5.14

试计算静态利差与绝对利差。

首先根据公司债券计算到期收益率。

```
>> r1 = irr([-103.7 8 8 8 8 108])
r1 =
    0.0710
```

得到公司债券到期收益率为 7.1 %。下面计算 5 年期国债到期收益率。考虑求解面值为 100 元的 5 年期国债票息，设票息为 c，则有下面方程：

$$100=\frac{c}{(1+4.51\ \%)}+\frac{c}{(1+4.68\ \%)^2}+\frac{c}{(1+4.84\ \%)^3}+\frac{c}{(1+4.99\ \%)^4}+\frac{c}{(1+5.14\ \%)^5}$$

解得 $c=5.109\ 4$。

找出对应到期收益率。

```
>> c = 5.1094;
```

```
>> r2 = irr([-100,c,c,c,c,100+c])
r2 =
    0.0511
```

这样5年期国债对应到期收益率为5.11 %,绝对利差为1.99 %(7.1 %−5.11 %=1.99 %)。下面求解静态利差。求静态利差实际上就是求解下面方程。

$$103.71=\frac{8}{(1+4.51\%+r_{ss})}+\frac{8}{(1+4.68\%+r_{ss})^2}+\frac{8}{(1+4.84\%+r_{ss})^3}+\frac{8}{(1+4.99\%+r_{ss})^4}+\frac{100+8}{(1+5.14\%+r_{ss})^5}$$

下面调用MATLAB中的函数进行求解。

```
>> eq = sym('103.7 = 8/(1 + 0.0451 + x) + 8/(1 + 0.0468 + x)^2 + 8/(1 + 0.0484 + x)^3... +
           8/(1 + 0.049 9 + x)^4 + (100 + 8)/(1 + 0.0514 + x)^5');
>> a = solve(eq,'x')
```

可以看到a的解有很多,包括复数根与负根,但只有一个($a=0.02$)符合要求,所以静态利差为0.02。

3.3.3 复杂形式现金流计算

1. 根据贴现率、债券发行日和到期日计算短期债券收益率

有些美国债券是以贴现的方式发行,例如,投资者以95元的价格买入面值为100元的短期债券,到期日获得100元,买入价与面值之间差就是投资者的收益。一般将贴现与面值的比率称为贴现率。

在MATLAB中根据贴现率、债券发行日以及到期日计算债券收益率的函数是tbilldisc2yield。

调用方式

```
[BEYield MMYield] = tbilldisc2yield(Discount, Settle, Maturity)
```

输入参数

Discount	贴现率
Settle	结算日
Maturity	到期日

输出参数

BEYield	根据一年365天计算的收益率
MMYield	根据一年360天计算的收益率

【例 3-19】 某债券结算日为 2002 年 10 月 1 日，到期日是 2003 年 3 月 31 日，年贴现率为 0.049 7，求债券收益率。

```
>> Discount = 0.0497;
>> Settle = '01-Oct-02';
>> Maturity = '31-Mar-03';
>> [BEYield MMYield] = tbilldisc2yield(Discount, Settle, Maturity)
BEYield =
          0.0517
MMYield =
          0.0510
```

2. 根据短期债券收益率计算贴现率

tbillyield2disc 是 tbilldisc2yield 函数的逆函数。

【例 3-20】 一种债券计息日是 2002 年 10 月 1 日，到期日是 2003 年 3 月 31 日，收益率是 4.97 %，求其贴现率。执行命令如下：

```
>> Yield = 0.0497;Settle = '01-Oct-02';Maturity = '31-Mar-03';
>> Discount = tbillyield2disc(Yield, Settle, Maturity)
Discount =
            0.0485
```

3. 计算债券价格

在 MATLAB 中根据债券收益率计算债券价格的函数是 tbillprice。

调用方式

```
Price = tbillprice(Rate, Settle, Maturity, Type)
```

输入参数

Rate	债券的年收益率
Settle	债券的结算日
Maturity	债券的到期日
Type	(Optional)债券的类型。Type = 1(默认值)表示货币市场工具，Type = 2 表示债券，Type = 3 表示贴现率。货币市场应计天数法则按 Act/360 法则计算，债券按 Act/365 计算，贴现率按 Act/360 计算

输出参数

Price	债券的价格

【例3-21】 已知债券结算日是2002年10月1日，到期日是2003年3月1日，债券收益率为4.5%，求该债券价格。这时可以执行如下命令：

```
>> Rate = 0.045;Settle = '01-Oct-02';Maturity = '31-Mar-03';
>> Price = tbillprice(Rate, Settle, Maturity, Type)
Price =
     97.8172
```

4. 根据零息债券价格计算收益率

当债券价格给定时，计算零息券到期收益率的是 zeroyield 函数。

调用方式

```
Yield = zeroyield(Price, Settle, Maturity, Period, Basis, EndMonthRule)
```

输入参数

Price	面值为100的债券价格，如果是可赎回的债券，也要将赎回价设为100
Settle	债券的结算日
Maturity	债券的到期日
Period	(Optional)年发放票息的频率
Basis	(Optional)应计天数的计算方式。内容如下： 0 = Act/Act (default) 1 = 30/360(SIA) 2 = Act/360 3 = Act/365 4 = 30/360 (PSA) 5 = 30/360 (ISDA) 6 = 30/360 (European) 7 = Act/365 (Japanese)
EndMonthRule	(Optional)月末法则。仅对到期日是30日或者小于30日有效，0表示发放票息的日期相同，1(默认值)表示票息在该月最后一天发放

输出参数

Yield	债券到期收益率

在MATLAB中零息券到期收益率分如下两种形式。如果是零息券或者只有一个似息票期(如果债券除了零息外还有其他支付方式)的赎回期，收益率计算公式如下：

$$\text{Yield} = \left(\frac{\text{RV} - P}{P}\right)\left(\frac{M \cdot E}{\text{DSR}}\right)$$

式中：前面一项是投资收益，后面一项是年度利率。

如果赎回期前有多个支付票息期间，则应采用下列公式计算：

$$\text{Yield} = \left[\left(\frac{\text{RV}}{P}\right)^{\frac{1}{N_q - 1 + \frac{\text{DSC}}{E}}} - 1\right] \cdot M$$

式中：DSC 为从结算日到下一个似息票日的天数，DSR 为从结算日到赎回日的天数，E 为似息票期间，M 为一年中似息票的时间段，N_q 为在计息日到赎回日之间似息票的时间段，P 为面值为 100 美元的零息券的价格，RV 为赎回的价格，Yield 为持有债券到赎回日收益。

【例 3-22】 计算一个短期债券收益率，结算日为 1993 年 6 月 24 日，到期日为 1993 年 11 月 1 日，应计天数计算方法为 Act/Act，该债券价格为 95 元，求其收益率。

```
>> Settle = '24-Jun-1993';
>> Maturity = '1-Nov-1993';
>> Basis = 0;
>> Price = 95;
>> Yield = zeroyield(Price, Settle, Maturity, [], Basis)
Yield =
      0.1490
```

如果应计天数按照 30/360(SIA)，下面计算其到期收益率。

```
>> Settle = '24-Jun-1993';
>> Maturity = '1-Nov-1993';
>> Basis = 1;
>> Price = 95;
>> Yield = zeroyield(Price, Settle, Maturity, [], Basis)
Yield =
      0.1492
```

5. 根据零息券收益率计算价格

调用方式

```
Price = zeroprice(Yield, Settle, Maturity, Period, Basis, EndMonthRule)
```

输入参数

Yield	收益率
Settle	结算日
Maturity	到期日
Period	票息支付频率

Basis	应计天数法则
EndMonthRule	月末法则

输出参数

Price	零息券价格

如果是零息券或者只有一个似息票期间(如果债券除了零息外还有其他支付方式)的赎回期，则价格计算公式如下：

$$\text{Price}=\frac{\text{RV}}{1+\left(\frac{\text{DSR}}{E}\times\frac{Y}{M}\right)}$$

如果赎回期前有多个支付票息期间，则应采用下列公式计算：

$$\text{Price}=\frac{\text{RV}}{\left(1+\frac{Y}{M}\right)^{N_q-1+\frac{\text{DSC}}{E}}}$$

式中：DSC是从结算日到下一个似息票日的天数；DSR是从结算日到赎回日的天数；E是似息债券的天数；M是一年中似息票的时间段；N_q是在计息日到赎回日之间似息票的时间段；Price是面值100美元的零息券的价格；RV是赎回的价格；Y是持有债券到期的年收益率。

【例3-23】 已知债券结算日为1993年6月24日，到期日为1993年11月1日，收益率为0.04，应计天数法则为Act/Act，计算该债券的价格。

```
>> Settle = '24 - Jun - 1993';
>> Maturity = '1 - Nov - 1993';
>> Period = 2;
>> Basis = 0;
>> Yield = 0.04;
>> Price = zeroprice(Yield, Settle, Maturity, Period, Basis)
Price =
        98.6066
```

3.3.4 根据收益率计算短期债券价格

调用方式

```
Price = tbillprice(Rate, Settle, Maturity, Type)
```

输入参数

Rate	债券的收益率
Settle	债券的结算日

Maturity	债券的到期日
Type	(Optional)债券的类型 1(默认值)表示货币市场(Act/360) 2 表示债券(Act/365) 3 表示贴现率(Act/360)

输出参数

Price	债券的价格

【例 3－24】 短期美国国债利率为 4.5 %，结算日为 2002 年 10 月 1 日，到期日为 2003 年 3 月 31 日，求该债券价格。

```
>> Rate = 0.045;
>> Settle = '01 - Oct - 02';
>> Maturity = '31 - Mar - 03';
>> Type = 2;                    % 计息方法采用 Act/365 方法
>> Price = tbillprice(Rate, Settle, Maturity, Type)
Price =
      97.8172
```

实际上该债券利率采用 Act/365 方法，从 2002 年 10 月 1 日到 2003 年 3 月 31 日，实际天数为 181 天。

```
>> Rate = 0.045;
>> price = Face/[1 + Rate * 181/365]
price =
      97.8172
```

3.3.5 根据短期国债价格计算收益率

调用方式

```
[MMYield, BEYield, Discount] = tbillyield(Price, Settle, Maturity)
```

输入参数

Price	面值为 100 的国库券的价格
Settle	结算日
Maturity	到期日

输出参数

MMYield	货币市场的收益

```
BEYield          债券市场的收益
Discount         债券的贴现率
```

货币市场的收益率按照实际天数/360(Act/360)法则计算，债券市场收益率按照实际天数/365(Act/365)法则计算。贴现率按照实际天数/360(Act/360)法则计算。

【例 3-25】 已知债券的价格为 98.75，结算日为 2002 年 10 月 1 日，到期日为 2003 年 3 月 31 日，将其分别折算成货币市场收益率、债券市场收益率、贴现率。

```
>> Price = 98.75;
>> Settle = '01-Oct-02';
>> Maturity = '31-Mar-03';
>> [MMYield, BEYield, Discount] = tbillyield(Price, Settle,Maturity)
MMYield =
          0.0252
BEYield =
          0.0255
Discount =
          0.0249
```

计算得到折算成货币市场收益率为 0.025 2，债券市场收益率为 0.025 5，贴现率为 0.024 9。

3.3.6 短期债券回购的计算

记 T_1 为 2002 年 9 月 26 日到 2002 年 12 月 26 日天数，T_2 为 2002 年 10 月 26 日到 2002 年 12 月 26 日天数，T_1，T_2 之间关系示意图如图 3.1 所示。

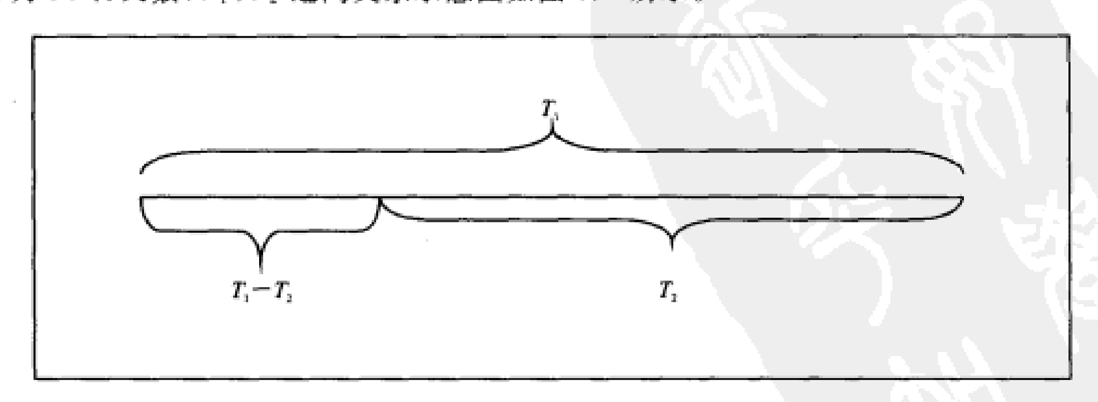


图 3.1 T_1，T_2 之间关系示意图

从图 3.1 中可以看出投资者可以有两种选择。一种是从 2002 年 9 月 26 日持有债券一直到期满；另一种是从 2002 年 9 月 26 日到 2002 年 10 月 26 日进行回购，然后从 2002 年 10 月 26 日到 2002 年 12 月 26 日投资其他品种债券。该债券贴现率应使得上述两种选择的最终收

益率相等。

调用方式

```
TBEDiscount = tbillrepo(RepoRate, InitialDiscount, PurchaseDate, SaleDate, Maturity)
```

输入参数

RepoRate	债券的回购率
InitialDiscount	初始贴现率
PurchaseDate	购买日期
SaleDate	卖出日期
Maturity	到期日

输出参数

TBEDiscount	回购盈亏平衡点的贴现率

【例3-26】 短期债券初始贴现率为0.047 5,债券到期日为2002年4月3日,购买债券日期为2002年1月3日,卖出日期为2002年2月3日,回购利率为0.045,求该项回购盈亏平衡点贴现率。

```
>> RepoRate = 0.045;
>> InitialDiscount = 0.0475;
>> PurchaseDate = '3-Jan-2002';
>> SaleDate = '3-Feb-2002';
>> Maturity = '3-Apr-2002';
>> TBEDiscount = tbillrepo(RepoRate,InitialDiscount,PurchaseDate,SaleDate, Maturity)
TBEDiscount =
            0.0491
```

【例3-27】 债券到期日为2002年12月26日,初始贴现率为1.61%,购买债券日期为2002年9月26日,卖出日为2002年10月26日,回购利率1.49%,计算回购盈亏平衡点贴现率。

```
>> Maturity = '26-Dec-2002';
>> InitialDiscount = 0.0161;
>> PurchaseDate = '26-Sep-2002';
>> SaleDate = '26-Oct-2002';
>> RepoRate = 0.0149;
>> BreakevenDiscount = tbillrepo(RepoRate, InitialDiscount, PurchaseDate, SaleDate,
Maturity)
BreakevenDiscount =
```

```
            0.0167
```

下面对回购过程的现金流进行计算。

第一步：计算购买日的短期债券的价格。

```
>> PriceOnPurchaseDate = tbillprice(InitialDiscount, PurchaseDate, Maturity, 3)
PriceOnPurchaseDate =
              99.5930
```

第二步：计算回购日的收益，应计天数采用30/360。

```
>> RepoInterest = RepoRate * PriceOnPurchaseDate * days360(PurchaseDate,SaleDate)/360
RepoInterest =
            0.1237
```

回购收益为0.123 7。

最后计算10月26日短期债券的价格。

```
>> PriceOnSaleDate = tbillprice(BreakevenDiscount, SaleDate,Maturity, 3)
PriceOnSaleDate =
            99.7167
```

表3.10列举了自回购以来的现金流。

表3.10 自回购以来的现金流

日 期	现金流出	金额/元	现金流入	金额/元
2002年9月26日	购买 T-bill	99.593	回购	99.593
2002年10月26日	赎回 T-bill	99.593	卖出 T-bill	99.716 8
	回购收益	0.123 8		
	合计	199.309 8		199.309 8

3.3.7 可转让定期存单应计利息

1. 可转让定期存单应计利息

在MATLAB中计算可转换定期存单应计利息的函数为cdai，面值默认100元。应计天数法则默认Act/360(SIA)。

调用方式

```
AccrInt = cdai(CouponRate, Settle, Maturity, IssueDate, Basis)
```

输入参数

CouponRate	票息率
Settle	结算日
Maturity	到期日
IssueDate	发行日
Basis	(Optional) 应计天数法则 2 表示 Act/360（默认值） 3 表示 = Act/365

输出参数

AccrInt	可转让定期存单的应计利息

【例 3-28】 已知可转让定期存单票息率为 0.05，结算日为 2002 年 1 月 2 日，到期日为 2002 年 3 月 31 日，发行日为 2001 年 10 月 1 日，求其应计利息。

```
>> CouponRate = 0.05;
>> Settle = '02-Jan-02';
>> Maturity = '31-Mar-02';
>> IssueDate = '1-Oct-01';
>> AccrInt = cdai(CouponRate, Settle, Maturity, IssueDate)
AccrInt =
        1.2917
```

可转让存单 CD(Certificate of Deposit)应计利息为 1.291 7。

2. 计算 CD 收益率

可转让定期存单收益率。

调用方式

```
Yield = cdyield(Price, CouponRate, Settle, Maturity, IssueDate, Basis)
```

输入参数

Price	面值是 100 美元的净价(Clean Price)。如果是全价(Dirty Price)可以用函数 cdai 计算应计利息
CouponRate	每年的收益率
Settle	结算日
Maturity	到期日
IssueDate	发行日
Basis	(Optional)应计天数法则

```
2 = Act/360(默认值)
3 = Act/365
```

输出参数

```
Yield            CD 的收益率
```

【例 3-29】 可转让定期存单(CD)的发行日是 2002 年 10 月 1 日，结算日是 2002 年 1 月 2 日，到期日是 2002 年 3 月 31 日，交易价格为 101.125 美元，求该可转让定期存单的收益率。图 3.2 是可转让存单时间的示意图。

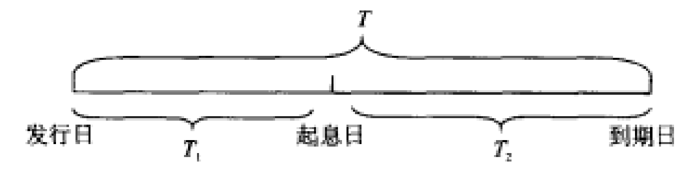


图 3.2 可转让存单时间的示意图

$$起息日的票息=票息\times T_1$$

$$起息日收益=\frac{结算价}{100}+起息日的票息$$

则有

$$可转让定期存单的收益=\left(\frac{1+票息收益率\times T}{起息日收益}-1\right)\times T_2$$

实际上根据公式有

```
>> Price = 101.125;
>> T1 = 0.2583;                 % 将 T1 折算成年
>> T2 = 0.2444;
>> T = 0.5027;
>> AccrInt = CouponRate * T1;
>> B = Price/100 + AccrInt;
>> Yield = ( (1 + CouponRate * T) / B - 1 ) * (1/T2)
Yield =
     0.0039
```

下面调用 cdyield 函数可以得到同样结果。

```
>> Price = 101.125;
>> CouponRate = 0.05;
>> Settle = '02 - Jan - 02';
>> Maturity = '31 - Mar - 02';
>> IssueDate = '1 - Oct - 01';
```

```
>> Yield = cdyield(Price, CouponRate, Settle, Maturity, IssueDate)
Yield =
      0.0039
```

3. 可转让存单 CD 价格

调用方式

```
[Price, AccrInt] = cdprice(Yield, CouponRate, Settle, Maturity, IssueDate, Basis)
```

输入参数

Yield	到期收益率
CouponRate	票息率
Settle, .	结算日
Maturity	到期日
IssueDate	发行日
Basis	(Optional)应计天数计算法则 2 表示 Act/360(默认值) 3 表示 Act/365

输出参数

Price	CD 净价(Clean Price),面值为 100 元
AccrInt	CD 应计利息

【例 3-30】 CD 收益率为 0.052 5,票息率 5 %,结算日为 2002 年 1 月 2 日,到期日为 2002 年 3 月 31 日,发行日为 2001 年 10 月 1 日。求 CD 价格和应计利息。

```
>> Yield = 0.0525;
>> CouponRate = 0.05;
>> Settle = '02-Jan-02';
>> Maturity = '31-Mar-02';
>> IssueDate = '1-Oct-01';
>> [Price, AccruedInt] = cdprice(Yield, CouponRate, Settle, Maturity, IssueDate)
Price =
      99.9233
AccruedInt =
            1.2917
```

该 CD 价格为 99.923 3,应计利息为 1.291 7。

3.3.8 长期债券到期收益率

调用方式

```
Yield = bndyield(Price, CouponRate, Settle, Maturity, Period, Basis, EndMonthRule,
        IssueDate, FirstCouponDate, LastCouponDate, StartDate, Face)
```

输入参数

Price	债券的净价(clean price),可以为一个向量
CouponRate	票息率
Settle	结算日
Maturity	到期日
Period	(Optional)每年支付票息的频率
Basis	(Optional)应计天数法则
EndMonthRule	(Optional)月末法则
IssueDate	(Optional)发行日
FirstCouponDate	(Optional)首次付息日
LastCouponDate	(Optional)最后一次付息日
StartDate	(Optional)现金收到日
Face	(Optional)债券的面值,默认值为100

输出参数

Yield	到期收益率

【例3-31】 已知债券的票息率为0.05,结算日为1997年1月20日,到期日为2002年6月15日,每年支付2次票息,应计天数法则为Act/Act,求到期收益率。

```
>> Price = 95;
>> CouponRate = 0.05;
>> Settle = '20-Jan-1997';
>> Maturity = '15-Jun-2002';
>> Period = 2;
>> Basis = 0;
>> Yield = bndyield(Price, CouponRate, Settle, Maturity, Period, Basis)
Yield =
           0.0610
```

债券到期收益率为0.061 0。

3.3.9　根据长期债券到期收益率计算净价

调用方式

```
[Price, AccruedInt] = bndprice(Yield, CouponRate, Settle, Maturity)
[Price, AccruedInt] = bndprice(Yield, CouponRate, Settle, Maturity, Period, Basis,…
    EndMonthRule, IssueDate, FirstCouponDate, LastCouponDate, StartDate, Face)
```

输入参数同上。

输出参数

Price	债券净价
AccruedInt	债券的应计收益

【例3-32】 已知债券以收益率报价，3种债券的收益率报价分别为0.04，0.05，0.06，票息率都为5 %，结算日为1997年1月20日，到期日为2002年6月15日，每年支付2次票息，应计天数采用Act/Act，价格为95元，求其净价。

```
>> Yield = [0.04; 0.05; 0.06];
>> CouponRate = 0.05;
>> Settle = '20-Jan-1997';
>> Maturity = '15-Jun-2002';
>> Period = 2;
>> Basis = 0;
>>[Price, AccruedInt] = bndprice(Yield, CouponRate, Settle,Maturity, Period, Basis)
Price =
       104.8106
       99.9951
       95.4384
AccruedInt =
              0.4945
              0.4945
              0.4945
```

3.3.10　现金流转换为对应债券

在MATLAB中将现金流转换为对应债券的函数是cfamounts。

调用方式

```
[CFlowAmounts, CFlowDates, TFactors, CFlowFlags] = cfamounts(CouponRate, Settle, Maturity, Pe-
```

riod, Basis, EndMonthRule, IssueDate, FirstCouponDate, LastCouponDate, StartDate, Face)

输入参数

CouponRate	债券票息
Settle	结算日
Maturity	到期日
Period	息票支付的频率
Basis	(Optional) 应计天数法则
EndMonthRule	(Optional) 月末法则
IssueDate	(Optional) 债券发行日
FirstCouponDate	(Optional) 债券息票首付日
LastCouponDate	(Optional) 最后一次付息日
StartDate	(Optional) 开始日期
Face	(Optional) 面值

输出参数

CFlowAmounts　　现金流的数量

CFlowDates　　现金流的日期

TFactors　　时间因子(time factors)。时间因子的计算需要求解下面的方程:

$$PV=\frac{CF}{\left(1+\frac{z}{2}\right)^{TF}}$$

式中：PV为现金流现值,CF为现金流,z为经过风险调整年收益率,TF是时间因子。

CFlowFlags　　现金流类型,其内容如下:

0　到期日的累计利息

1　初次现金流,发放的日期一般小于正常的票息期间(coupon period,相邻两次票息间隔的时间)

2　初次现金流,发放时间大于正常的票息期间

3　到期现金流为名义票息

4　到期日现金流为本金加利息

5　距离到期日不满一个票息期间,最后一次票息期间短于其他票息期间,到期现金流小于名义现金流

6　由于最后一个票息期间长于其他票息期间,导致到期现金流高于正常的到期现金流

7　距离到期日不满一个票息期间,现金流为到期现金流

8　距离到期日不满一个票息期间,现金流小于到期日现金流

9　距离到期日不满一个票息期间,现金流大于到期日现金流

10 零息券的到期现金流

【例 3-33】 将两组现金流转换为对应的债券，现金流的结算日为 1993 年 11 月 1 日，到期日为 1994 年 12 月 15 日与 1995 年 6 月 15 日，票息率分别为 6 %与 5 %，债券发放票息次数分别为 4 次和 2 次。应计天数计算规则分别是 Act/Act 与 30/360(SIA)，试给出对应债券。

```
>> Settle = '01-Nov-1993';
>> Maturity = ['15-Dec-1994';'15-Jun-1995'];
>> CouponRate = [0.06; 0.05];
>> Period = [4; 2];
>> Basis = [1; 0];
>> [CFlowAmounts, CFlowDates, TFactors, CFlowFlags] = cfamounts(CouponRate,Settle, Maturity,
   Period, Basis)
CFlowAmounts =
                -0.7667     1.5000     1.5000     1.5000     1.5000     101.5000
                -1.8989     2.5000     2.5000     2.5000   102.5000        NaN
CFlowDates =
             728234       728278       728368       728460       728552       728643
             728234       728278       728460       728643       728825        NaN
TFactors =
           0      0.2404     0.7403     1.2404     1.7403     2.2404
           0      0.2404     1.2404     2.2404     3.2404        NaN
CFlowFlags =
             0      3      3      3      3      4
             0      3      3      3      4    NaN
>> datestr(CFlowDates(1,1:6))
ans =
      01-Nov-1993
      15-Dec-1993
      15-Mar-1994
      15-Jun-1994
      15-Sep-1994
      15-Dec-1994
>> datestr(CFlowDates(2,1:5))
ans =
      01-Nov-1993
      15-Dec-1993
      15-Jun-1994
```

```
15 - Dec - 1994
15 - Jun - 1995
```

表 3.11 分别是两组现金流所对应债券。

表 3.11 现金流转化为债券格式

日 期	1993-11-1	1993-12-15	1994-3-15	1994-6-15	1994-9-15	1994-12-15
债券 1	1.5	1.5	1.5	1.5	1.5	101.5
债券 2	2.5	2.5	2.5	2.5	102.5	NaN

3.3.11 可转换债券定价

可转换债券是指可转换为股票的债券，兼有公司债券和股票双重特点。一般情况下事先规定票面利率、转股价、转股比率和转换期。票面利率就是债券持有到期时收益率；转股价是指可转换债券在转股期间转化为基础股票的每股价格；转换比率是指发行转债的公司向投资者约定比率，按照这一比率，投资者可以将手中债券转换为相应数量股票；转换期是指可转换债券转换为股票起始日至结束日的期间，通常有下面 4 种情况：

- 发行后某日至到期前某日；
- 发行后某日至到期日；
- 发行日至到期日前某日；
- 发行日至到期日。

在前两种情况下，可转换债券有一段时间锁定期，这段时间内债券持有人不可以将之转换为股票。

可转换债券相当于一份债券或者股票，如果没有换成股票则相当于持有债券，如果转成股票就相当于持有股票，所以可转换债券的价格可视为债券部分与股票部分之和。

在 MATLAB 中采用了 Cox - Ross - Rubinstein 二叉树的方法来计算可转换债券价格。

调用方式

```
[CBMatrix, UndMatrix, DebtMatrix, EqtyMatrix] = cbprice(RiskFreeRate, StaticSpread, Sigma,
Price, ConvRatio, NumSteps, IssueDate, Settle, Maturity, CouponRate, Period, Basis, EndMonthRule,
DividendType, DividendInfo, CallType, CallInfo, TreeType)
```

输入参数

RiskFreeRate	无风险利率
StaticSpread	静态利差，为超过无风险利率部分
Sigma	股票波动的标准差
Price	标的资产的价格

ConvRatio	转换比例
NumSteps	计算的步数
IssueDate	发行日
Settle	结算日
Maturity	到期日
CouponRate	票息率
Period	二叉树离散的时间段
Basis	(Optional)应计天数法则
EndMonthRule	(Optional)月末法则,每月最后一日的处理方式。当月末等于30日或者小于30日时该法则有效,0表示忽略该法则,1表示票息的支付总是在月末。默认值为1
DividendType	(Optional)票息类型,0(默认值)表示红利以绝对形式发放,1以收益率形式表示
DividendInfo	(Optional)除息信息,默认值表示没有除息。第一行是除息日期,第二行是除息的数量
CallType	(Optional)看涨期权的类型,0(默认值)表示现金价,1表示净价(clean price),不含应计利息的价格
CallInfo	(Optional)第一列为日期,面值为100元的债券的回购价格,默认值为无赎回条款
TreeType	(Optional)树形图的类型,0(默认值)表示二叉树,1表示三叉树

输出参数

CBMatrix	可转换债券的可转换价格矩阵,CBMatrix(1,1)为可转换价格
UndMatrix	二叉树格式的债券价格
DebtMatrix	可转换债券价格的债券部分,为一矩阵形式
EqtyMatrix	可转换债券价格的股票部分,为一矩阵形式

【例3-34】 已知无风险利率为0.05,标的资产波动标准差为0.3,可转换债券转换比率为1∶1,二叉树时间离散数目为200个时间段。债券发行日为2002年1月2日,结算日为2002年1月2日,到期日为2007年1月2日,票息率为0.04,每年支付2次利率,应计天数采用1=30/360 (SIA)方法。月末法则默认值为1,红利以绝对数量美元支付(而非比率),在2004年1月2日以110元的价格赎回。

```
RiskFreeRate = 0.05;
Sigma        = 0.3;
ConvRatio    = 1;
NumSteps     = 200;
IssueDate    = datenum('2-Jan-2002');
Settle       = datenum('2-Jan-2002');
Maturity     = datenum('2-Jan-2007');
```

```
CouponRate    = 0.04;
Period        = 2;
Basis         = 1;
EndMonthRule = 1;
DividendType = 0;
DividendInfo = [];
CallInfo      = [datenum('2-Jan-2004'), 110];
CallType      = 1;
TreeType      = 1;
```

下面计算可转换债券的价格，考虑了不同的静态利差以及债券的价格，以观察对利率期限结构的影响。

```
for j = 0:0.005:0.015;                %4 种不同的静态利差水平
StaticSpread = j;
      for i = 0:10:100
          Price = 40 + i;             %债券的不同价格
          [CbMatrix, UndMatrix, DebtMatrix, EqtyMatrix] = ...
          cbprice(RiskFreeRate, StaticSpread, Sigma, Price, ...
          ConvRatio, NumSteps, IssueDate, Settle, ...
          Maturity, CouponRate, Period, Basis, EndMonthRule, ...
          DividendType, DividendInfo, CallType, CallInfo, ...
          TreeType);
          convprice(i/10 + 1,j * 200 + 1) =   CbMatrix(1,1);
          stock(i/10 + 1,j * 200 + 1)       =   Price;
        end
end
    plot(stock, convprice);
    legend({'+0 bp'; '+50 bp'; '+100 bp'; '+150 bp'});
    title ('根据三叉树方法计算的有效价差 - 200 步')
    xlabel('证券价格');
    ylabel('可转换债券价格');
    text(50, 150, ['票息为 4% 每半年支付一次.', sprintf('\n'), ...
          '2 年后按全价为 110 赎回.' sprintf('\n'), ...
          '5 年后到期.'],'fontweight','Bold')
```

图 3.3 是可转换债券价格与股票价格关系示意图。

下面就静态利差等于 0.015、债权价格为 140 时验证可转债的债券部分与股票部分的价格。

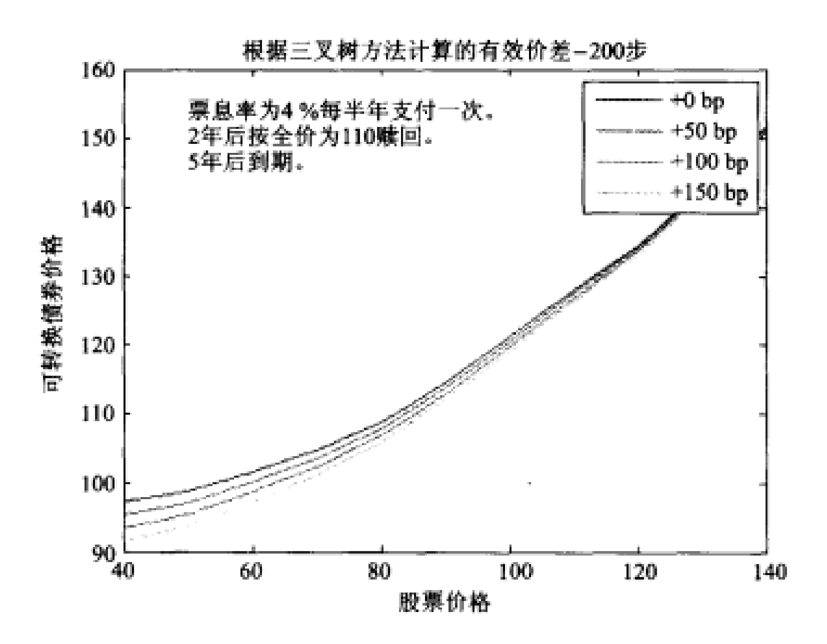


图 3.3 可转换债券价格与股票价格关系图

```
>> price = 140;
>> StaticSpread = 0.015
>> [CbMatrix, UndMatrix, DebtMatrix, EqtyMatrix] = ...
        cbprice(RiskFreeRate, StaticSpread, Sigma, Price, ...
        ConvRatio, NumSteps, IssueDate, Settle, ...
        Maturity, CouponRate, Period, Basis, EndMonthRule, ...
        DividendType, DividendInfo, CallType, CallInfo, ...
        TreeType);
>> CbMatrix(1,1)          % 可转换债券价格
ans =
     150.8532
>> DebtMatrix(1,1)        % 债券部分的价格
ans =
     16.5449
>> EqtyMatrix(1,1)        % 股票部分的价格
ans =
     134.3083
>> DebtMatrix(1,1) + EqtyMatrix(1,1)
ans =
     150.8532
```

可以看出可转换债券价格等于股票部分价格与债券部分价格之和，因此可转换债券可以视为股票与债券两种产品复合体。

3.3.12 固定收益久期与凸度

久期(duration)是金融学一个重要概念，由佛雷德里希·麦考利(Federich Macaulay)和约翰·西克斯爵士(John Hicks)大致在同一时间发现的，当时麦考利的目标是找出一个可以用来比较期限相同但是支付结构不同债券的方法，西克斯的目标是衡量债券对于利率风险敞口。

麦考利久期(MD)

$$\mathrm{MD} = \frac{\mathrm{d}P}{\mathrm{d}r}\frac{1+r}{P}$$

在 MATLAB 中计算久期的函数是 cfdur，计算公式如下：

$$D = \frac{\mathrm{PV}(t_1)t_1 + \mathrm{PV}(t_2)t_2 + \cdots + \mathrm{PV}(t_n)t_n}{\mathrm{PV}}$$

式中：$\mathrm{PV}(t_i)$表示 t_i 时期现值。

业内人士通常使用的是修正久期(modified duration)。修正久期是指当收益率改变时债券的预期现金流不变情况下，收益率变化 1 %，债券价格变化百分比。修正久期计算公式如下：

$$\text{修正久期} = \frac{\text{麦考利久期}}{1+\dfrac{\text{债券到期收益率}}{k}}$$

式中：k 表示每年支付利息次数。

调用方式

```
[Duration, ModDuration] = cfdur (CashFlow, Rate)
```

输入参数

CashFlow	为各期现金流
Rate	贴现率

输出参数

Duration	久期
ModDuration	修正久期

【例 3-35】 一项投资各期现金流如表 3.12 所列。

表 3.12　各期投资现金流

期　限	第一期	第二期	第三期	第四期	第五期
金额/元	1 000	2 000	3 000	4 000	5 000

贴现率为 0.025，问该项投资久期是多少？

首先将各期现金流保存到变量 CashFlow，然后调用 cfdur 函数。输入命令如下：

```
>> CashFlow = [1000 2000 3000 4000 5000]
CashFlow =
        1000        2000        3000        4000        5000
>> [Duration, ModDuration] = cfdur(CashFlow, 0.025)
Duration =
        3.6279
ModDuration =
              3.5394
```

从上面可以看出久期是 8.970 9，修正久期是 8.752 1。

久期本质上是价值曲线在当前利率和债券价格点斜率，凸性则是斜率的变化量，债券价格 P 随利率 r 的变化而变化，习惯上就可以把债券价格视为利率函数，利用泰勒级数展开得到

$$\Delta P \approx \frac{\mathrm{d}P}{\mathrm{d}r}\Delta r + \frac{1}{2}\frac{\mathrm{d}^2 P}{\mathrm{d}r^2}(\Delta r)^2 \tag{3.1}$$

式(3.1)两边同除以价格 P，则上式变成

$$\frac{\Delta P}{P} \approx \frac{\frac{\mathrm{d}P}{\mathrm{d}r}}{P}\Delta r + \frac{1}{2}\frac{\frac{\mathrm{d}^2 P}{\mathrm{d}r^2}}{P}(\Delta r)^2 \tag{3.2}$$

这样定义凸度 C 如下：

$$C = \frac{\frac{\mathrm{d}^2 P}{\mathrm{d}r^2}}{P} \tag{3.3}$$

则式(3.2)左边为债券价格变化率，右边关于 Δr 的一次项、二次项系数分别是久期与凸度。式(3.2)可以理解如下：

债券价格变化＝－久期×债券价格×债券收益率的变化＋$\frac{1}{2}$凸度×债券价格×收益率变化^2

下面根据现金流贴现公式推导出债券凸度。设债券将来各期现金流分别为 $C_1, C_2, \cdots, C_T$，相应时间为 $t_1, t_2, t_3, \cdots, t_T$，债券贴现率为 r，由现值公式得出债券价格表达式为

$$P = \sum_{t=t_1}^{T} \frac{C_t}{(1+r)^t}$$

债券凸度为

$$C = \frac{1}{P}\frac{d^2P}{dr^2} = \frac{1}{P(1+r)^2}\left[\sum_{t=t_1}^{T} t(1+t)\frac{C_t}{(1+r)^t}\right]$$

可以证明麦考利久期与凸度之间关系为

$$C = \frac{1}{(1+r)^2}[S + \mathrm{MD}(\mathrm{MD}+1)]$$

式中：$S = \sum_{t=t_1}^{T} \frac{c_t(1+r)^{-t}}{P}(t-\mathrm{MD})$，MD是麦考利久期，$S$是衡量现金流集中程度。当久期给定时，现金流越集中，则债券凸度越大，现金流越分散，债券凸度越小。

在MATLAB中计算凸度的函数有很多种，计算现金流凸度的函数为cfconv。

调用方式

```
Convexity = cfconv(CashFlow, Yield)
```

输入参数

```
CashFlow        各期现金流
Yield           收益率
```

输出参数

```
Convexity       凸度
```

【例3-36】 某金融产品现金流为连续9年支付2.5，第10年还本付息，求该现金流凸度。

```
>> CashFlow = [2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 102.5];
>> Convex = cfconv(CashFlow, 0.025)
Convex =
        90.4493
```

该现金流凸度为90.449 3。

3.4 利率期限结构

利率期限结构是未来利率水平的重要指标。利率期限结构代表市场未来利率预期，投资者通过利率期限曲线形状推测未来利率走势。解读利率期限结构既是一门科学更是一门艺术；债券投资者、基金经理和企业财务人员都会认真研究利率期限结构；中央银行、财政部也需要解读利率期限结构，结合通货膨胀率等其他信息来制定国家利率政策。

3.4.1 计算利率期限结构

利率期限结构理论主要讨论金融资产到期时收益与到期期限之间关系。假设某一投资者

打算作 N 年长期投资，他可以有两种选择：一种是购买在 N 年底到期的长期债券，并且持有直到期满为止；另外一种方法是先持有1年期债券，到期后再购买1年期债券，如此不断直到 N 年期满为止。

实际上，任何债券都可以视为零息券组合，付息债券可以视为一系列到期现金流的组合。债券价格就是这些现金流现值。如要对每一期现金流进行贴现，需要知道贴现率，一般选和该现金流日期相同的零息券，这一收益率称为即期利率（spot rate）。但是众所周知零息券大都为短期债券，所以一般构建国债即期收益率曲线（spot rate curve），首先要选择以何种国债收益率曲线为基础；可供选择的国债有新发行国债、部分非新发行国债、所有附息中长期国债和短期国债、零息券等。

1. 步步为营法计算零息率

在MATLAB中采用步步为营法（boot straping）计算利率期限结构，如计算3年期利率期限结构，假设各期收益率如表3.13所列。

表3.13 平价收益率

年	0.5	1.0	1.5	2.0	2.5	3.0
到期收益率	4.0	4.2	4.6	5.6	5.8	7.0

一年以内债券采用折价销售，一年以上债券每半年支付一次利息，收益率为票息率，面值为100美元，债券票息利率就是到期收益率。

下面给出1.5年的国库券现金流。

0.5年：$0.046\times100\times0.5=2.3$

1.0年：$0.046\times100\times0.5=2.3$

1.5年：$0.046\times100\times0.5+100=102.3$

现金流量现值为

$$\frac{2.3}{\left(1+\frac{r_1}{2}\right)}+\frac{2.3}{\left(1+\frac{r_2}{2}\right)^2}+\frac{102.3}{\left(1+\frac{r_3}{2}\right)^3}$$

式中：r_1 为第0.5年即期利率；r_2 为第1.0年即期利率；r_3 为第1.5年即期利率。

由于1年以内的债券为零息券，所以 $r_1=0.04$，$r_2=0.042$。根据现金流现值等于面值得下列方程

$$100=\frac{2.3}{\left(1+\frac{r_1}{2}\right)}+\frac{2.3}{\left(1+\frac{r_2}{2}\right)^2}+\frac{102.3}{\left(1+\frac{r_3}{2}\right)^3}$$

求解得 $r_3=4.62$。

有了 r_3 就可以求解第2年的即期利率。

2 年期国债现金流为

0.5 年：$0.056\times100\times0.5=2.8$

1.0 年：$0.056\times100\times0.5=2.8$

1.5 年：$0.056\times100\times0.5=2.8$

2.0 年：$0.056\times100\times0.5+100=102.8$

其现金流现值为

$$\frac{2.8}{\left(1+\frac{r_1}{2}\right)}+\frac{2.8}{\left(1+\frac{r_2}{2}\right)^2}+\frac{2.8}{\left(1+\frac{r_3}{2}\right)^3}+\frac{102.8}{\left(1+\frac{r_4}{2}\right)^4}$$

根据现金流现值等于面值 100，可得出下列方程

$$\frac{2.8}{\left(1+\frac{r_1}{2}\right)}+\frac{2.8}{\left(1+\frac{r_2}{2}\right)^2}+\frac{2.8}{\left(1+\frac{r_3}{2}\right)^3}+\frac{102.8}{\left(1+\frac{r_4}{2}\right)^4}=100$$

由于 r_1，r_2，r_3 都是已知，根据上面方程求解得 $r_4=5.66$。以此类推可以计算出各时点上的即期利率，具体如表 3.14 所列。

表 3.14　各时点即期利率

年	0.5	1.0	1.5	2.0	2.5	3.0
到期收益率	4.0	4.2	4.62	5.66	5.86	7.18

在 MATLAB 中计算利率时间期限结构的函数为 zbtyield。

调用方式

```
[ZeroRates, CurveDates] = zbtyield(Bonds, Yields, Settle,OutputCompounding)
```

输入参数

Bonds	票息的时间、票息、面值
Yields	每种债券的到期收益率向量
Settle	结算日
OutputCompounding	期限结构的票息支付方式，可以取下面的值 1　每年支付 1 次票息 2　（默认值）每半年支付 1 次票息 3　每 4 个月支付 1 次票息 4　每季度支付 1 次票息 6　每 2 月支付 1 次票息 12　每月支付 1 次票息 -1　表示连续复利

输出参数

CurveDates 曲线上的日期

ZeroRates 零息率曲线日期对应的利率

注意本书的Compounding参数如无特别说明，其内容同本例。

【例3-37】 已知国债面值是100美元，各期收益率如表3.15所列。

表3.15 国债各期收益

短期国债品种	票 息	到期日	到期收益率
3个月	0	17-Apr-2003	1.15
6个月	0	17-Jul-2003	1.18
2年	1.75	31-Dec-2004	1.68
5年	3.00	15-Nov-2007	2.97
10年	4.00	15-Nov-2012	4.01
30年	5.375	15-Feb-2031	4.92

下面分析由上述品种构成零息率曲线。

```
>> %首先需要输入债券特征
>> Bonds = [datenum('04/17/2003')    0        100;
            datenum('07/17/2003')    0        100;
            datenum('12/31/2004')    0.0175   100;
            datenum('11/15/2007')    0.03     100;
            datenum('11/15/2012')    0.04     100;
            datenum('02/15/2031')    0.05375  100];
>> Yields = [0.0115;
             0.0118;
             0.0168;
             0.0297;
             0.0401;
             0.0492];
>> Settle = datenum('17-Jan-2003');          %输入结算日
>> [ZeroRates, CurveDates] = zbtyield(Bonds, Yields, Settle)
ZeroRates =
           0.0115
           0.0118
           0.0168
```

```
            0.0302
            0.0418
            0.0550
CurveDates =
             731688
             731779
             732312
             733361
             735188
             741854
>> datestr(CurveDates)              % 将天数转换为日期
ans =
      17 - Apr - 2003
      17 - Jul - 2003
      31 - Dec - 2004
      15 - Nov - 2007
      15 - Nov - 2012
      15 - Feb - 2031
```

利率期限结构如表 3.16 所列。

表 3.16 国债利率期限结构

日　期	2003-4-17	2003-7-17	2004-12-31	2007-11-5	2012-11-15	2031-2-15
利　率	0.011 5	0.011 8	0.016 8	0.030 2	0.041 8	0.055 0

进一步地画出利率期限结构示意图。

```
>> plot(CurveDates,ZeroRates)
>> datetick('x',2)              % 设定坐标轴 X 的刻度为日期型
```

利率期限结构曲线结果如图 3.4 所示。

2. 根据债券价格求零息率曲线

调用方式

```
[ZeroRates, CurveDates] = zbtprice(Bonds, Prices, Settle, OutputCompounding)
```

输入参数

Bonds　　Bonds 为一个向量，分别为债券的日期、票息、票面价值、期间、日期计算方法及月末法则

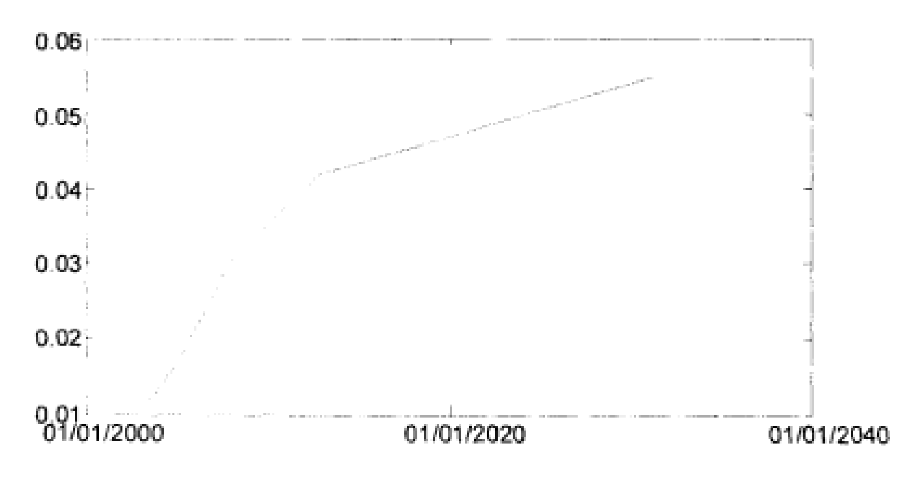


图 3.4　利率期限结构示意图

Prices	债券的当前价格
Settle	结算日期
OutputCompounding	(Optional)期限结构利息的支付方式

输出参数

ZeroRates	各个日期的利率
CurveDates	与利率对应的日期

【例 3-38】 给定债券的到期日、票息率、票面价值及价格如表 3.17 所列。

表 3.17　债券的到期日、票息率、票面价值及结算日等

到期日	票息率	面　值	价　格	付息次数	日期方式	月末法则
6/1/1998	0.04 75	100	99.375	2	0	0
7/1/2000	0.06	100	99.875	2	0	0
7/1/2000	0.093 75	100	105.75	6	1	0
6/30/2001	0.051 25	100	96.875	1	3	1
4/15/2002	0.071 25	100	101.125	4	0	0
1/15/2000	0.08	100	103.125	2	3	0
9/1/1999	0.058 75	100	99.375	3	0	0
4/30/2001	0.071 25	100	101	2	0	0
11/15/1999	0.07	100	101.25	2	3	1
7/1/2001	0.052 5	100	96.375	2	3	0
4/30/2002	0.07	100	102.75	2	0	0

债券结算日为 12/18/1997，利率按半年复利计算，求零息率。首先给出各债券特征如下：

```
>> Bonds = [datenum('6/1/1998')     0.0475     100   2   0   0;
            datenum('7/1/2000')     0.06       100   2   0   0;
            datenum('7/1/2000')     0.09375    100   6   1   0;
            datenum('6/30/2001')    0.05125    100   1   3   1;
            datenum('4/15/2002')    0.07125    100   4   1   0;
            datenum('1/15/2000')    0.065      100   2   0   0;
            datenum('9/1/1999')     0.08       100   3   3   0;
            datenum('4/30/2001')    0.05875    100   2   0   0;
            datenum('11/15/1999')   0.07125    100   2   0   0;
            datenum('6/30/2000')    0.07       100   2   3   1;
            datenum('7/1/2001')     0.0525     100   2   3   0;
            datenum('4/30/2002')    0.07       100   2   0   0];
```

债券价格保存在 Price 变量中。

```
>> Prices = [99.375;
              99.875;
             105.75;
              96.875;
             103.625;
             101.125;
             103.125;
              99.375;
             101.0;
             101.25;
              96.375;
             102.75 ];
>> Settle = datenum('12/18/1997')
Settle =
          729742
>> OutputCompounding = 2;
>> [ZeroRates, CurveDates] = zbtprice(Bonds, Prices, Settle,OutputCompounding)
ZeroRates =
            0.0616
            0.0609
            0.0658
            0.0590
```

```
            0.0648
            0.0655
            0.0606
            0.0601
            0.0642
            0.0621
            0.0627
CurveDates =
             729907
             730364
             730439
             730500
             730667
             730668
             730971
             731032
             731033
             731321
             731336
>> plot(CurveDates,ZeroRates)
>> datetick('x',2)              % 设定坐标轴 X 的刻度为日期型
```

根据价格计算出的利率期限结构图如图 3.5 所示。

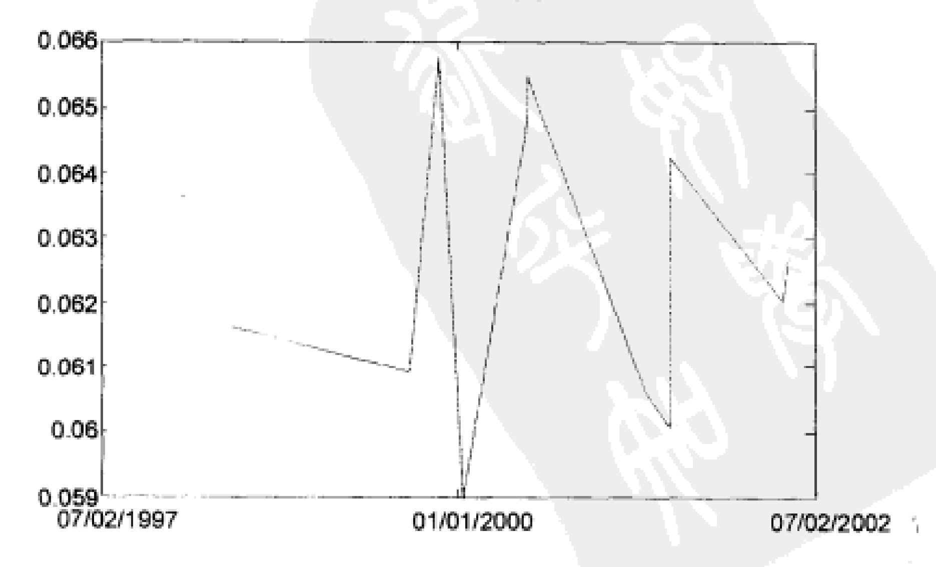


图 3.5　根据价格计算出的利率期限结构图

从图3.5可以看出利率并不光滑，需要进一步拟合使其变得光滑。

3.4.2 拟合利率期限结构

有时利率期限结构不够光滑，这时需要用对其进行平滑处理，但是一平滑就会产生误差，拟合目的就要在光滑与精度之间进行取舍，使得期限结构曲线在光滑的同时保持较高精度。MATLAB中的termfit函数利用三次样条函数对期限结构进行拟合。

调用方式

```
[ZeroRates, CurveDates, BootZeros, BootDates, BreakDates] = termfit(Smoothing, Bonds,
    Prices, Settle, OutputCompounding, OutputBasis, CurveDates, BreakDates)
```

输入参数

Smoothing	平滑性控制系数，在0到1之间。0保证最小拟合误差，1保证曲线最光滑
Bonds	债券包，为一 $N \times M$ 矩阵，N 表示债券的品种，M 表示债券的特征。第一列为到期日，第二列为票息率，第三列为面值(默认值为100)，第四列为票息支付频率，为1、2(默认值)、3、4、6、12。第五列为Basis，应计天数计算法则，内容如下： 0 - Act/Act(默认值) 1 - 30/360 2 - Act/360 3 - Act/365 第六列是月末法则
Prices	各个债券的价格向量
Settle	利率期限结构的起始日
OutputCompounding	(Optional)拟合利率每年支付利息频率，默认值是2
OutputBasis	(Optional)拟合利率应计天数计算法则，内容如下： 0 - Act/Act(默认值) 1 - 30/360 2 - Act/360 3 - Act/365
CurveDates	(Optional)需要拟合的日期，为日期行向量
BreakDates	(Optional)间断点日期，为日期行向量。在结算日和最长债券到期日之间

输出参数

ZeroRates	拟合曲线各点日期对应的利率
CurveDates	拟合曲线各点日期
BootDates	利率期限结构的日期
BootZeros	自举法得出的利率期限结构日期对应的利率
BreakDates	间断点对应的日期

【例 3-39】 对【例 3-38】的利率曲线进行拟合。

```
>> Smoothing = 0.5
>> Bonds = [datenum('6/1/1998')    0.0475    100    2    0    0;
            datenum('7/1/2000')    0.06      100    2    0    0;
            datenum('7/1/2000')    0.09375   100    6    1    0;
            datenum('6/30/2001')   0.05125   100    1    3    1;
            datenum('4/15/2002')   0.07125   100    4    1    0;
            datenum('1/15/2000')   0.065     100    2    0    0;
            datenum('9/1/1999')    0.08      100    3    3    0;
            datenum('4/30/2001')   0.05875   100    2    0    0;
            datenum('11/15/1999')  0.07125   100    2    0    0;
            datenum('6/30/2000')   0.07      100    2    3    1;
            datenum('7/1/2001')    0.0525    100    2    3    0;
            datenum('4/30/2002')   0.07      100    2    0    0];
```

债券价格保存在 Prices 变量中。

```
>> Prices = [99.375;
             99.875;
            105.75;
             96.875;
            103.625;
            101.125;
            103.125;
             99.375;
            101.0;
            101.25;
             96.375;
            102.75];
>> Settle = datenum('12/18/1997');
>> OutputCompounding = 2;
>> [ZeroRates, CurveDates, BootZeros, BootDates, BreakDates]...
= termfit(Smoothing, Bonds, Prices, Settle,OutputCompounding)
```

下面画出利率曲线拟合图。

```
>> plot(CurveDates,ZeroRates,'b-', BootDates,BootZeros,'b*')
>> datetick('x',2)
```

利率期限结构拟合结果如图 3.6 所示。

图 3.6 利率期限结构以及拟合图

图 3.6 中的点是自举法得出的利率，光滑曲线是拟合的结果。

3.4.3 计算远期利率

通常在市场中有很多种债券，每种都有发行日与到期日，有时也需要考虑将来某个时间段利率，例如 2000 年 2 月 15 日财政部发行分别于 2000 年 8 月 15 日、2001 年 2 月 15 日、2001 年 8 月 15 日、2002 年 2 月 15 日以及 2002 年 8 月 15 日到期 5 种国债。如果投资者在 2002 年 2 月到 2002 年 8 月计划对流动资金进行融资，那他就需要关注 2002 年 2 月 15 日到 2002 年 8 月 15 日利率期限结构。

1. 远期利率的计算方法

从理论上讲，有了利率期限结构就可以推导出任意时刻的远期利率。为了计算的方便，引入下面的记号。用 f 代表远期，确定远期利率时必须提供两个数据：一是远期利率开始的时间；二是远期利率的时间段。引入记号${}_nf_t$，表示自 n 时刻起 t 时间段后的远期利率。

如果利率期限结构的每个时间段是半年，则 5 年相当于 10 个半年，2 年相当于 4 个半年，这样 5 年后的 2 年隐含远期利率记为${}_{10}f_4$；类似地 3 年后的 7 年期隐含远期利率记为${}_6f_{14}$。3 年期后 2 年时间隐含远期利率记为${}_6f_4$，表 3.18 是利率期限结构，每半年为一个时间段，r_t 表示 t 期的即期利率。下面计算 3 年期后 2 年时间段的远期利率${}_6f_4$。

表 3.18　利率期限结构

利　率	期限/年	即期利率/年	即期利率/半年
1	0.5	0.08	$r_1=0.04$
2	1.0	0.083	$r_2=0.041\ 5$
3	1.5	0.089 3	$r_3=0.044\ 65$
4	2.0	0.092 47	$r_4=0.046\ 235$
5	2.5	0.094 68	$r_5=0.047\ 34$
6	3.0	0.097 87	$r_6=0.048\ 935$
7	3.5	0.101 29	$r_7=0.050\ 645$
8	4.0	0.105 29	$r_8=0.052\ 645$
9	4.5	0.108 5	$r_9=0.054\ 25$
10	5.0	0.110 21	$r_{10}=0.055\ 105$

首先假设一位准备投资 5 年期国债的人，他可以有如下两种方法。

方法 1：直接购买 5 年期(10 个半年)国债。

方法 2：先购买 3 年期(6 个半年)国债，同时在 3 年期满后购买 2 年期(4 个半年)国债。

方法 1 与方法 2 的 5 年后现金流终值是相同的，所以他的收益率应该是相等的，即

$$(1+r_{10})^{10}=(1+r_6)^6(1+{}_6f_4)^4$$

式中：r_6、r_{10} 分别为第 6 期(3 年)、第 10 期(5 年)的即期利率，${}_6f_4$ 为第 3 年开始的 2 年期远期，求解得

$${}_6f_4=\left[\frac{(1+r_{10})^{10}}{(1+r_6)^6}\right]^{1/4}-1$$

带入方程得

$${}_6f_4=\left[\frac{(1+1.055\ 105)^{10}}{(1+1.048\ 935)^6}\right]^{1/4}-1=0.064\ 4$$

注意，0.064 4 是半年期的收益率，折算成年的远期利率为 12.88 %(0.064 4×2)。

2. 远期利率的计算函数

在 MATLAB 中 ratetimes 函数可以计算远期利率。函数 ratetimes 输入参数分成两个部分：第一部分是说明原始利率期限结构；第二部分是给出新的日期。输出结果是对应于新日期的利率结构。

计算远期利率期限结构，调用方式根据时间格式不同有两种。

调用方式

```
Rates = ratetimes(Compounding, RefRates, RefEndTimes, RefStartTimes, EndTimes, StartTimes)
Rates = ratetimes(Compounding, RefRates, RefEndDates, RefStartDates, EndDates, StartDates, Val-
            uationDate)
```

输入参数

Compounding	每年支付票息的频率,可以选取 1,2,3,4,6,12,365 等。
RefRates	利率期限结构日期对应的利率
RefEndTimes	利率期限结构的结束的时间
RefStartTimes	利率期限结构开始的时间
EndTimes	远期结束的时间
StartTimes	远期开始的时间
RefEndDates	利率期限结构结束的日期
RefStartDates	利率期限结构开始的日期
EndDates	远期结束日期
StartDates	远期开始日期
ValuationDate	远期的评估日

输出参数

Rates	对应于新日期的利率结构

【例 3-40】 如果已知原始利率期限结构如下。

开始日	结束日	利率
15 Feb 2000	15 Aug 2000	0.05
15 Feb 2000	15 Feb 2001	0.056
15 Feb 2000	15 Aug 2001	0.06
15 Feb 2000	15 Feb 2002	0.065
15 Feb 2000	15 Aug 2002	0.075

现在需要研究对应于下面日期的利率。

开始日	结束日	利率
15 Feb 2000	15 Aug 2000	?
15 Aug 2000	15 Feb 2001	?
15 Feb 2001	15 Aug 2001	?
15 Aug 2001	15 Feb 2002	?
15 Feb 2002	15 Aug 2002	?

可以按照如下步骤进行。

首先输入原始利率期限结构。

```
>> RefStartDates = ['15 - Feb - 2000'];
>> RefEndDates = ['15 - Aug - 2000'; '15 - Feb - 2001'; '15 - Aug - 2001';'15 - Feb - 2002';…'15 - Aug - 2002'];
>> Compounding = 2;
>> ValuationDate = ['15 - Feb - 2000'];
>> RefRates = [0.05; 0.056; 0.06; 0.065; 0.075];
```

然后输入新时间段。

```
>> StartDates = ['15 - Feb - 2000'; '15 - Aug - 2000'; '15 - Feb - 2001';'15 - Aug - 2001'; '15 - Feb - 2002'];
>> EndDates = ['15 - Aug - 2000'; '15 - Feb - 2001'; '15 - Aug - 2001';'15 - Feb - 2002'; '15 - Aug - 2002'];
```

调用 ratetimes 函数计算对应于新时间段的利率。

```
>> Rates = ratetimes(Compounding, RefRates, RefEndDates, RefStartDates, EndDates,…, StartDates, ValuationDate)
```

结果如下：

```
Rates =
    0.0500
    0.0620
    0.0680
    0.0801
    0.1155
```

需要注意的是 MATLAB 的远期利率期限结构是线性插值法得出的，所以不够平滑，感兴趣的读者可以用样条插值来提高光滑度。

思考题

1. 了解常用的计息天数计算方法。
2. 编写程序计算固定收益久期与凸度。
3. 表 3.19 是固定收益的到期收益率，面值都为 100。

表 3.19 固定收益到期收益率

短期国债	票 息	到期日	到期收益率
3 个月	0	17 - Apr - 2003	1.13
6 个月	0	17 - Jul - 2003	1.17
2 年	1.75	31 - Dec - 2004	1.58
5 年	3.00	15 - Nov - 2007	2.67
10 年	4.00	15 - Nov - 2012	4.01
30 年	5.375	15 - Feb - 2031	4.92

计算利率期限结构示意图然后进行拟合。

4. 本章中静态利差可以用牛顿迭代法进行数值求解，试编写出程序。

5. 试分析国债回购利率与上证指数之间相关系数。



第4章 资产组合计算

资产组合是实务性比较强的内容。本章要求读者熟悉资产组合基本理论，掌握协方差与相关系数之间的相互推导，学会计算均值与方差，重点掌握资产组合有效前沿计算，能够处理无风险利率以及借贷关系情况下的最优投资组合，会用 MATLAB 规划工具箱求解非线性多约束条件下的最优投资组合问题。

4.1 资产组合基本原理

证券投资组合理论(portflio theory)主要研究如何配置各种不同金融资产，满足投资者的各种要求。1952 年美国学者马克维茨(H. Markowitz)创立资产组合理论，该理论在实践中得到广泛运用。

4.1.1 协方差矩阵与相关系数矩阵转换

在 MATLAB 中 corr2cov 函数把相关系数矩阵转换为协方差矩阵。

调用方式

```
Covariances = corr2cov(STDs, Correlations)
```

输入参数

```
STDs            各个资产的标准差组成的向量
Correlations    相关系数矩阵
```

输出参数

```
Covariances     协方差矩阵
```

【例 4-1】 已知资产组合中有 3 个品种，各资产预期回报、标准差及相关系数如表 4.1 所列。

表 4.1 各资产预期回报、标准差及相关系数

项 目	资产 A	资产 B	资产 C
预期回报	0.1	0.15	0.12
标准差	0.2	0.25	0.18

续表 4.1

项目		资产 A	资产 B	资产 C
相关系数矩阵	资产 A	1	0.8	0.4
	资产 B	0.8	1	0.3
	资产 C	0.4	0.3	1

计算资产协方差矩阵。

```
>> Returns = [0.1 0.15 0.12];
>> STDs = [0.2 0.25 0.18];
>> Correlations = [ 1     0.8   0.4
                    0.8   1     0.3
                    0.4   0.3   1 ];
>> Covariances = corr2cov(STDs, Correlations)
Covariances =
              0.0400      0.0400      0.0144
              0.0400      0.0625      0.0135
              0.0144      0.0135      0.0324
```

4.1.2 资产组合收益率与方差

在 MATLAB 中计算资产组合的回报与方差的函数是 portstats。

调用方式

```
[PortRisk, PortReturn] = portstats(ExpReturn, ExpCovariance, PortWts)
```

输入参数

ExpReturn	各资产期望收益向量
ExpCovariance	资产的协方差矩阵
PortWts	资产权重向量

输出参数

PortRisk	总资产的标准差
PortReturn	总资产的收益

【例 4-2】 两个资产组合中有 3 种资产 A、B、C，组合中各资产预期回报、协方差矩阵及权重如表 4.2 所列。

表4.2 资产组合中各资产的明细表

项目		资产A	资产B	资产C
协方差	资产A	0.010 0	−0.006 1	0.004 2
	资产B	−0.006 1	0.040 0	−0.025 2
	资产C	0.004 2	−0.025 2	0.022 5
预期回报		0.1	0.2	0.15
组合1各资产权重		0.4	0.2	0.4
组合2各资产权重		0.2	0.4	0.4

```
>> ExpReturn = [0.1 0.2 0.15];
>> ExpCovariance = [0.0100     - 0.0061     0.0042
                   - 0.0061     0.0400    - 0.0252
                     0.0042    - 0.0252     0.0225 ];
>> PortWts = [0.4 0.2 0.4; 0.2 0.4 0.4];
>> [PortRisk, PortReturn] = portstats(ExpReturn, ExpCovariance,PortWts)
PortRisk =
          0.0560
          0.0451
PortReturn =
            0.1400
            0.1600
```

从上述结果可以看到这两个资产组合标准差分别为0.056、0.045 1，资产回报分别为0.14、0.16。

【例4-3】 假设资产组合中有5种资产，收益分别为0.1、0.12、0.14、0.16、0.2，方差为0.02、0.03、0.01、0.05、0.02，资产收益率各不相关，各资产权重分别为0.1、0.2、0.3、0.2、0.2，计算该组合收益率与方差。

```
>> returns = [0.1 0.12 0.14 0.16 0.2];variances = [0.02 0.03 0.01 0.05 0.02];
>> ws = [0.1 0.2 0.3 0.2 0.2];
>> mean = dot(returns,ws)
mean =
        0.1480
>> variance = sum(variances. * ws.^2)
variance =
```

```
0.0051
```

该资产组合收益率为0.148 0，方差为0.005 1。

4.1.3 资产组合

VaR(Value-at-Risk)即在险价值，按照Philippe Jorion的定义是指在正常的市场环境下，给定一定的置信水平，预期未来一段时间内的损失。定义c为置信水平，X为资产组合的损失，则一定持有期T内的Var(为正值，表示损失的绝对额)定义公式为

$$\mathrm{Prob}(X \leqslant -\mathrm{Var}) = 1 - c$$

式中：Var值即为一定置信水平下，资产组合在未来T期内的最大损失。

【例4-4】 假设投资者拥有两种资产，资产组合总价值为1 000万元，各资产权重分别为1/4与3/4，这两个资产日波动率的均值分别为0.003，0.002，标准差分别为0.02，0.01，这两个资产之间相关系数为0.8。置信水平为1 %，求该资产组合未来10天的Var。

首先求总资产方差，公式如下：

$$\bar{\sigma}^2 = [w_1\sigma_1 \quad w_2\sigma_2]\begin{bmatrix}1 & \rho \\ \rho & 1\end{bmatrix}\begin{bmatrix}w_1\sigma_1 \\ w_2\sigma_2\end{bmatrix} \tag{4.1}$$

式中：w_1，w_2分别为资产组合权重，σ_1，σ_2为各资产标准差，ρ为这两种资产之间相关系数。

一般地，可将上式用向量与矩阵形式表示，记$\boldsymbol{w}=[w_1 \quad w_2 \cdots w_n]$表示各个资产权重，$\boldsymbol{\sigma}=[\sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_n]$表示各个资产标准差，资产协方差矩阵记为cov，则式(4.1)可以改写为如下形式：

$$\bar{\sigma}^2 = (w \cdot \sigma)\mathrm{cov}(w,\sigma)^{\tau} \tag{4.2}$$

式中："·"为向量点乘符号，表示两个向量对应元素乘积，在MATLAB中用记号".*"表示向量点乘。记号τ表示向量转置。如果记$\sigma=w\cdot\sigma$，δt为时间，则有

$$\bar{\sigma}^2 = \sigma \cdot \mathrm{cov} \cdot \sigma^{\tau} \tag{4.3}$$

有了资产组合方差，就可以计算出Var数值，从正态分布表中可以查到对应于置信度水平$\alpha=0.01$的$Z_{1-\alpha}=2.326\ 3$。在各个资产都是服从正态分布假设下，资产Var值为

$$\mathrm{Var} = 总资产 \times (\mu \times \delta t - Z_{1-\alpha} \times \sqrt{\delta t} \times \bar{\sigma}) \tag{4.4}$$

具体来讲计算Var步骤如下所述。

第一步：输入资产权重向量$\boldsymbol{w}$、各资产标准差sigma以及资产之间相关系数cov，注意协方差矩阵一定是对称矩阵，需要计算的时间长度为δt。

第二步：权重向量点乘标准差向量。

第三步：计算资产总的标准差$\bar{\sigma}$。

第四步：对于给定置信水平α，查正态分布表找到$Z_{1-\alpha}$。

第五步：计算Var，Var=总资产$\times(\mu\times\delta t-Z_{1-\alpha}\times\sqrt{\delta t}\times\bar{\sigma})$。

在 Command 窗口下执行命令：

```
>> w = [1/4 3/4];                                %w 为权重向量
>> ret = [0.003 0.002];                          %ret 为各资产回报
>> sigma = [0.02 0.01];                          %sigma 为资产方差
>> corrcoef = [1,0.8;0.8,1];                     %corrcoef 为相关系数矩阵
>> deltat = 10;                                  %deltat 为时间长度
>> pret = deltat * dot(w,ret)                    %资产组合回报
>> sig = w. * sigma;
>> tsig = sig * cov * sig' * deltat;
>> var = 10^7 * (pret - 2.3263 * sqrt(tsig))     %计算 Var 值
var =
     - 6.4930e + 005
```

10 天 Var 值为 649 300 元。

实际上在 MATLAB 中有专门计算 Var 值的 portvrisk 函数，注意输入总资产期望收益与标准差，而不是组合中各个资产预期收益率与标准差。

调用方式

```
ValueAtRisk = portvrisk(PortReturn, PortRisk, RiskThreshold, PortValue)
```

输入参数

PortReturn	总资产的回报
PortRisk	总资产的标准差
RiskThreshold	概率阈值，默认值为 0.05
PortValue	资产总的价值

输出参数

ValueAtRisk	概率阈值下的单资产 Var 值

【例 4－5】 已知资产组合年回报率为 0.002 9，标准差为 0.030 8，资产现在价值为 1 亿元，求 1 %水平下资产组合未来 1 年的在险价值。

```
>> PortReturn = 0.0029;            %资产组合的收益
>> PortRisk = 0.0308;              %资产组合的标准差
>> RiskThreshold = 0.01;           %1%水平下损失的概率
>> PortValue = 1;
>> ValueAtRisk = portvrisk(PortReturn,PortRisk,RiskThreshold,PortValue)
ValueAtRisk =
            0.068 8
```

该资产 Var 值等于 0.068 8,即该资产未来 1 年损失 0.068 8 亿元的可能性为 1 %。

4.2 投资组合评价指标

4.2.1 夏普比率

夏普比率又被称为夏普指数(Sharp ratio),由诺贝尔奖获得者威廉·夏普于 1966 年最早提出,目前已成为衡量基金绩效最为常用的一个指标。如果夏普比率为正值,则说明该基金的平均净值增长率超过了无风险利率。如果同期银行存款利率作为无风险利率,则说明基金比银行存款要好。夏普比率越大,说明基金单位风险所获得的风险回报越高。夏普比率为负时,按大小排序没有意义。

以夏普比率的大小对基金表现加以排序的理论基础在于,假设投资者可以以无风险利率进行借贷,这样,通过确定适当的融资比例,高夏普比率的基金总是能够在同等风险的情况下获得比低夏普比率基金高的投资收益。例如,假设有两个基金 A 和 B,A 基金的年平均净值增长率为 20 %,标准差为 10 %,B 基金的年平均净值增长率为 15 %,标准差为 5 %,年平均无风险利率为 5 %,那么,基金 A 和基金 B 的夏普比率分别为 1.5 和 2,依据夏普比率,基金 B 的风险调整收益要好于基金 A。为了更清楚地对此加以解释,可以以无风险利率的水平,融入等量的资金(融资比例为 1∶1),投资于 B。那么,B 的标准差将会扩大 1 倍,达到与 A 相同的水平,但这时 B 的净值增长率则等于 25 %(即 2×15 %−5 %)则要大于 A 基金。

夏普比率的计算公式为

$$\text{sharpe} = \frac{\bar{r} - r_f}{\sigma}$$

式中:sharpe 为夏普比率,$\bar{r}$为资产的平均收益,r_f 为无风险利率,σ 为资产收益率的标准差。

调用方式

```
Ratio = sharpe(Asset, Cash)
```

输入参数

Asset	资产收益率序列
Cash	(Optional) 无风险资产的收益率序列,默认值为 0

输出参数

Ratio	夏普比率

在 MATLAB 自带的数据文件 FundMarketCash 中含有某共同基金 5 年的月数据,其中变量 TestData 每列的数据分别为共同基金价格、市场价格以及 90 天国库券的月价格,变量

Assets保存了数据的名称。

```
>> load FundMarketCash
%将价格序列转化为收益率序列
>> Returns = tick2ret(TestData);
%计算夏普比率
>> sha = sharpe(Returns(:,1),Returns(:,3))
sha =
      0.0886
```

4.2.2 信息比率

与夏普比例从绝对收益和总风险角度来描述不同，信息比率(information ratio)是从主动管理的角度描述风险调整后收益，信息比率的定义如下：

$$\text{Inforatio} = \frac{E(R - R_{\text{Benchmark}})}{\sigma(R - R_{\text{Benchmark}})}$$

式中：Inforatio 为信息比率，R 为资产收益率序列，$R_{\text{Benchmark}}$ 为基准收益率序列，$R - R_{\text{Benchmark}}$ 为超额收益率，$E(R - R_{\text{Benchmark}})$ 与 $\sigma(R - R_{\text{Benchmark}})$ 分别为超额收益均值与标准差。

调用方式

```
Ratio = Inforatio(Asset, Benchmark)
```

输入参数

Asset	资产的收益率序列
Benchmark	基准资产的收益率序列

输出参数

Ratio	信息比率

下面选用 FundMarketCash 文件中的数据，第二列市场收益率为基准收益率，计算投资组合信息比率。

```
>> load FundMarketCash
>> Returns = TestData(:,1);Benchmark = TestData(:,2);
>> InfoRatio = inforatio(Returns, Benchmark)
InfoRatio =
          3.1123
```

4.3 资产组合最大跌幅

4.3.1 历史最大跌幅

在金融计算中，一般假设资产的运动是布朗运动。布朗运动又可分为算术布朗运动和几何布朗运动两种。

算术布朗运动模型如下：

$$dX(t) = \mu dt + \sigma dW_t$$

几何布朗运动模型如下：

$$dS(t) = \mu_0 S(t)dt + \sigma_0 S(t)dW_t$$

在 MATLAB 中计算历史最大跌幅的函数是 maxdrawdown。

调用方式

```
MaxDD = maxdrawdown(Data)
MaxDD = maxdrawdown(Data, Format)
[MaxDD, MaxDDIndex] = maxdrawdown(Data, Format)
```

输入参数

Data	资产价格矩阵，每列为一种资产
Format	计算模型及方法，取值如下：
	'return'(默认值) 根据从最高点与最低点的落差计算
	'arithmetic' 根据算术平均模型计算最大下跌幅度
	'geometric' 根据几何平均模型计算最大下跌幅度

输出参数

MaxDD	最大跌幅
MaxDDIndex	最大跌幅起止时间，第一行分别为最大跌幅的起始时间，第二行分别为结束时间

下面用 MATLAB 自带的数据计算最大跌幅。

```
>> load FundMarketCash
>> plot(TestData(:,1))                % 绘出共同基金的市值图
```

下面画出共同基金价格图，其波段最高点与最低点如图 4.1 所示。

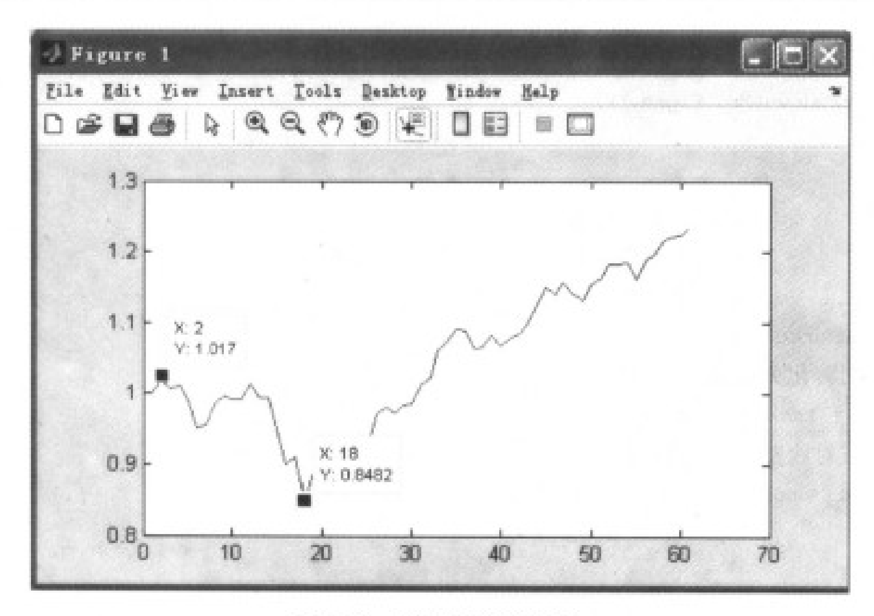


图 4.1 共同基金价格图

下面计算波段最大跌幅。

```
>> 0.8482/1.0168 - 1
ans =
      - 0.1658
>> maxdrawdown(TestData(:,1),'return')
ans =
      0.1658
% 根据算术模型计算波段最大跌幅
>> maxdrawdown(TestData(:,1),'arithmetic')
ans =
      0.1686
% 根据几何平均方法计算波段最大跌幅
>> log(0.8482/1.0168)
ans =
      - 0.1813
>> maxdrawdown(TestData(:,1),'geometric')
ans =
      0.1813
```

4.3.2 预期最大跌幅

假设标的资产服从几何布朗运动，$dX(t)=\mu dt+\sigma dW_t$。在 MATLAB 中计算预期最大跌幅的函数是 emaxdrawdown。

调用方式

```
EDD = emaxdrawdown(Mu, Sigma, T)
```

输入参数

```
Mu                  正态分布的均值
Sigma               正态分布的标准差
T                   选取的时间长度
>> load FundMarketCash
%将价格序列转化为收益率序列
>> Returns = tick2ret(TestData);
%用正态分布对收益率进行拟合
>> [mean std] = normfit(Returns(:,1))
mean =
      0.0038
std =
      0.0231
```

计算未来 3 个月的最大跌幅。

```
>> EDD = emaxdrawdown(mu, sigma, 3)
EDD =
      0.0458
```

4.4 资产组合有效前沿

由于证券市场投资存在巨大风险，一般不主张把投资集中在一种产品上，运用组合理论可以有效地降低投资风险，其核心思想是在目标收益率给定的情况下，要求资产组合风险最小。资产组合理论是由马克维茨(H. Markowitz)1952 年提出。

$$\min: \sigma^2 = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{x} = \sum_{i,j} x_i x_j \sigma_{i,j}$$

$$\text{s.t}\quad E(r_x) = \boldsymbol{x}^{\mathrm{T}}E(r) = \sum x_i E(r_i) = \mu;\ \sum x_i = 1$$

式中：μ 为投资者目标收益率，$\boldsymbol{V}$ 为协方差矩阵，$\boldsymbol{V}=(\sigma_{i,j})$，$r_i$ 为第 i 种资产收益率，x_i 为第 i 种资产在总资产中所占权重。

4.4.1 两种资产组合收益期望与方差

假设两种资产的收益率分别用 R_1，R_2 表示，协方差分别为 $\boldsymbol{V}=(\sigma_{i,j})$。记资产组合为 P，资产组合收益率、方差分别为 R_P 与 σ_P^2，x_1，x_2 分别表示权重，则有

$$R_P = x_1 R_1 + x_2 R_2$$

该资产组合期望收益率与方差为

$$E(R_P) = x_1 E(R_1) + x_2 E(R_2)$$

$$\sigma_P^2 = (x_1, x_2)\begin{pmatrix} \sigma_1^2 & \sigma_{1,2} \\ \sigma_{1,2} & \sigma_2^2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1^2\sigma_1^2 + 2x_1x_2\sigma_{1,2} + x_2^2\sigma_2^2$$

这样资产组合收益率均值与方差如图 4.2 所示。

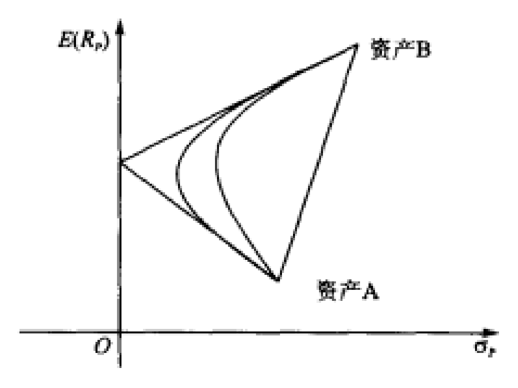


图 4.2　资产组合均值与方差

Markowitz 资产组合理论就是寻找一个有效组合，所谓有效组合是指在同样风险水平下具有最高收益，这样不同收益及与最小风险构成有效前沿。

在不允许卖空情况下，求解有效组合目标函数为

$$\min \frac{1}{2}\boldsymbol{x}\boldsymbol{V}\boldsymbol{x}^{\mathrm{T}}$$

$$\text{s.t}\quad \boldsymbol{x}^{\mathrm{T}}\overline{R} = \overline{R}_P;\qquad x_1 + x_2 = 1, x_1, x_2 \geqslant 0$$

这是一个约束条件为线性且含有不等式的二次规划方程，给定一个组合收益率就有一个最小方差，这样组合收益与最小方差构成有效前沿关系。资产组合有效前沿如图 4.3 所示。

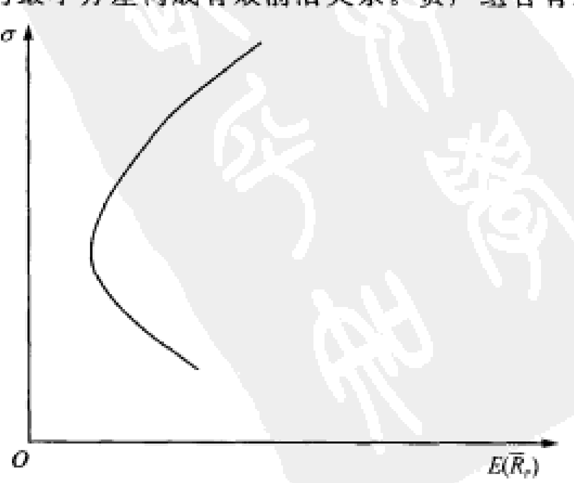


图 4.3　资产组合有效前沿示意图

4.4.2 均值方差有效前沿

在MATLAB工具箱中包含了资产均值方差有效前沿函数，这些都是基于MATLAB中的最优化理论工具箱。

调用方式

```
[PortRisk, PortReturn, PortWts] = frontcon(ExpReturn, ExpCovariance, NumPorts,PortReturn, Asset-
                                   Bounds, Groups, GroupBounds)
```

输入参数

ExpReturn	资产组合中每项资产预期回报，为一列行向量
ExpCovariance	各个资产之间协方差矩阵，为对称矩阵
NumPorts	(Optional)在资产组合有效前沿上的点的个数，默认值是10个点
PortReturn	(Optional)有效前沿上每个点回报
AssetBounds	(Optional)每种资产权重的上限、下限区间
Groups	(Optional)如果Groups(i,j)=1表示第i个资产属于第j个群，Groups(i,j)=0表示第i个资产不属于第j个群
GroupBounds	(Optional)每个群的权重约束区间，默认值规定下限为0，上限为1。

输出参数

PortRisk	组合的标准差
PortReturn	组合的回报
PortWts	组合中每个资产的权重

注意可以用“[]”代替默认值。

【例4-6】 考虑一个3资产组合，预期收益率分别为0.1、0.2、0.15，资产协方差矩阵如表4.3所列。求该资产组合有效前沿。

表4.3 资产协方差阵

项 目	资产1	资产2	资产3
资产1	0.01	−0.006 1	0.004 2
资产2	−0.006 1	0.04	−0.025 2
资产3	0.004 2	−0.025 2	0.022 5

```
>> ExpReturn = [0.1 0.2 0.15];
>> ExpCovariance = [ 0.0100     - 0.0061     0.0042
                    - 0.0061     0.0400     - 0.0252
                      0.0042    - 0.0252     0.0225];
```

```
>> NumPorts = 4;              % 资产组合有效前沿上选 4 个点
>> [PortRisk, PortReturn, PortWts] = frontcon(ExpReturn, ExpCovariance, NumPorts)
```

有效前沿上每个点的标准差、回报及各资产权重如下：

```
PortRisk =

        0.0426
        0.0483
        0.1089
        0.2000

PortReturn =

        0.1569
        0.1713
        0.1856
        0.2000

PortWts =

    0.2134    0.3518    0.4348
    0.0096    0.4352    0.5552
         0    0.7128    0.2872
         0    1.0000         0
```

4.4.3 带约束条件资产组合有效前沿

投资组合中的问题很少有简单的约束，大多数情况下是多种约束。监管当局为了防止流动性风险，对资产组合中每种资产的比例加以种种限制。例如，证监会规定封闭式基金持股上限为 80 %，开放式基金持股上限为 95 %，单只基金持有一家公司发行的股票，不得超过该基金资产净值的 10 %；同一基金管理公司管理的全部基金持有一家公司发行的股票，不得超过该公司总股本的 10 %。这时就需要考虑多个约束条件下的最优组合问题。

MATLAB 利用均值-方差理论求解资产组合问题，首先是将约束条件写成矩阵形式，例如 $\boldsymbol{Ax}\leqslant\boldsymbol{b}$ 或者 $\boldsymbol{Ax}=\boldsymbol{b}$ 形式。下面用一个例子说明。

【例 4-7】 某资产组合中有 5 种资产构成，第 i 种资产预期回报率为 $r_i=(i=1,2,3,4,5)$，w_i 为第 i 种资产在总资产中权重，考虑 w_i 具有如下形式：

$$0\leqslant w_1\leqslant 0.35;\quad 0\leqslant w_2\leqslant 0.3;\quad 0\leqslant w_3\leqslant 0.3$$

$$0\leqslant w_4\leqslant 0.4;\quad 0\leqslant w_5\leqslant 0.35$$

$$0.2\leqslant w_1+w_2\leqslant 0.6$$

$$0.3\leqslant w_3+w_4+w_5\leqslant 0.7$$

上述约束条件写成矩阵形式如下：

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0.35 \\ 0 & 1 & 0 & 0 & 0 & 0.30 \\ 0 & 0 & 1 & 0 & 0 & 0.30 \\ 0 & 0 & 0 & 1 & 0 & 0.40 \\ 0 & 0 & 0 & 0 & 1 & 0.5 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & 0 & -0.2 \\ 0 & 0 & -1 & -1 & -1 & -0.3 \\ 1 & 1 & 0 & 0 & 0 & 0.6 \\ 0 & 0 & 1 & 1 & 1 & 0.7 \end{pmatrix}$$

注意约束条件 $0 \leqslant w_1 \leqslant 0.35$ 可以分解成两个约束条件，$0 \leqslant w_1$（可以写成约束条件 $-w_1 \leqslant 0$）和 $w_1 \leqslant 0.35$，分别对应于矩阵的第 1 行、第 6 行。下面计算约束条件下资产组合有效前沿。

调用方式

```
[PortRisk, PortReturn, PortWts] = portopt(ExpReturn, ExpCovariance, NumPorts, PortReturn, ConSet)
```

输入参数

ExpReturn	资产的期望回报率
ExpCovariance	资产的协方差
NumPorts	(Optional)资产组合中投资品种的个数
PortReturn	(Optional)要求组合的回报率
ConSet	(Optional)约束条件

输出参数

PortRisk	资产组合的风险
PortReturn	资产组合的回报
PortWts	组合中各个资产的权重

【例 4-8】 设有两种资产其回报率分别为 0.1，0.3，协方差矩阵为 $\begin{pmatrix} 0.02 & 0 \\ 0 & 0.04 \end{pmatrix}$，要求组合没有借贷，且第一种资产的权重小于 20 %，第二种资产的权重大于 30 %，约束条件写成如

下形式：

$$w_1 + w_2 = 1$$
$$0 \leqslant w_1 \leqslant 0.2$$
$$0.3 \leqslant w_2$$

求该两种资产组合有效前沿。

```
>> ret = [0.1 0.3];cov = [0.02 0;0 0.04];
>> constr = [1 1 1;1 0 0.2; - 1 0 0 ;0 - 1 - 0.3];         % 约束矩阵
>> portopt(ret,cov,[],[],constr)
```

则满足上述条件的资产组合的均值方差有效前沿如图 4.4 所示。

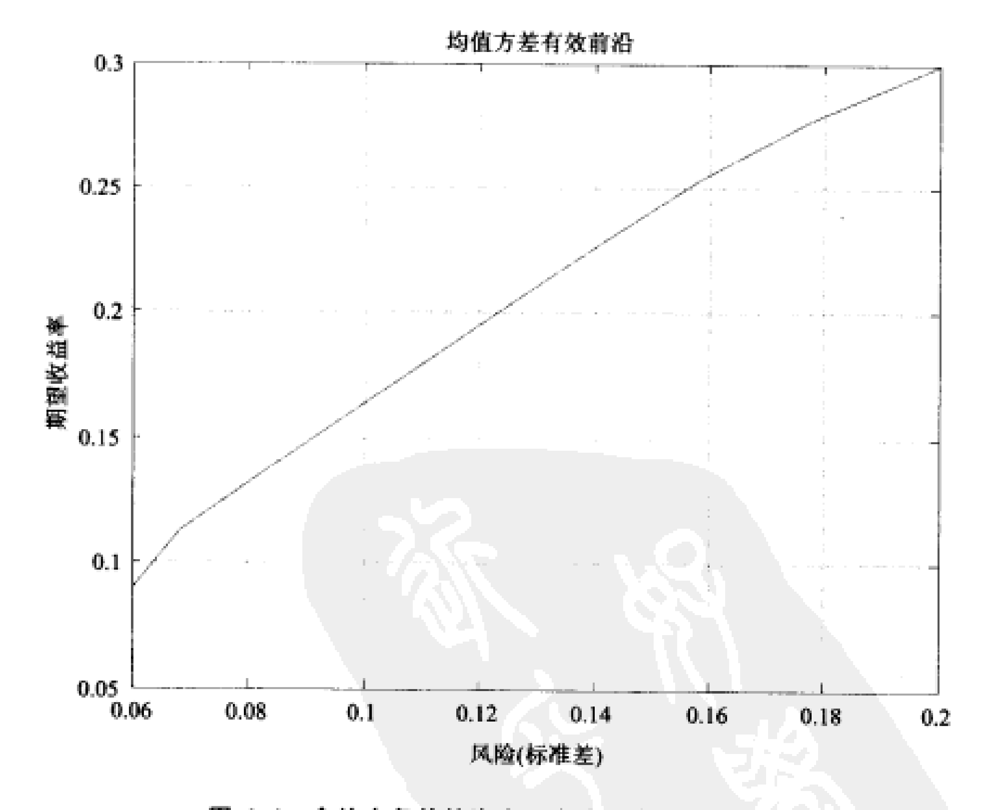


图 4.4　含约束条件的资产组合均值方差有效前沿

【例 4－9】 各资产相关系数矩阵、预期回报及标准差如表 4.4 所列。

表 4.4 各资产相关系数矩阵、预期回报及标准差

项目		资产 A	资产 B	资产 C
相关系数矩阵	资产 A	1	0.8	0.4
	资产 B	0.8	1	0.3
	资产 C	0.4	0.3	1
预期回报		0.1	0.15	0.12
各资产标准差		0.2	0.25	0.18

试计算有效前沿。

```
>> Returns       = [0.1 0.15 0.12];
>> STDs          = [0.2 0.25 0.18];
>> Correlations  = [ 1    0.8   0.4
                    0.8   1     0.3
                    0.4   0.3    1 ];
>> Covariances = corr2cov(STDs, Correlations);          % 相关系数转换为协方差矩阵
>> portopt(Returns, Covariances, 20)                    % 绘出组合的有效前沿
>> hold on
```

选择不同权重的资产组合。

```
>> rand('state', 0);
>> Weights = rand(1000, 3);                             % 产生 1000 行×3 列随机数
>> Total = sum(Weights, 2);                             % Weights 中数据沿对列求和
>> Weights(:,1) = Weights(:,1)./Total;
>> Weights(:,2) = Weights(:,2)./Total;
>> Weights(:,3) = Weights(:,3)./Total;                  % 这时 Weights 变成了权重矩阵
```

绘出各个资产组合风险与收益图。

```
>> [PortRisk, PortReturn] = portstats(Returns, Covariances, Weights);
>> plot (PortRisk, PortReturn, '.r')
>> title('均值-方差有效前沿以及各个资产组合风险与收益。')
>> xlabel('风  险(标准差)')
>> ylabel('期望收益率')
>> hold off
```

这样资产组合有效前沿和各个资产组合风险与收益如图 4.5 所示。

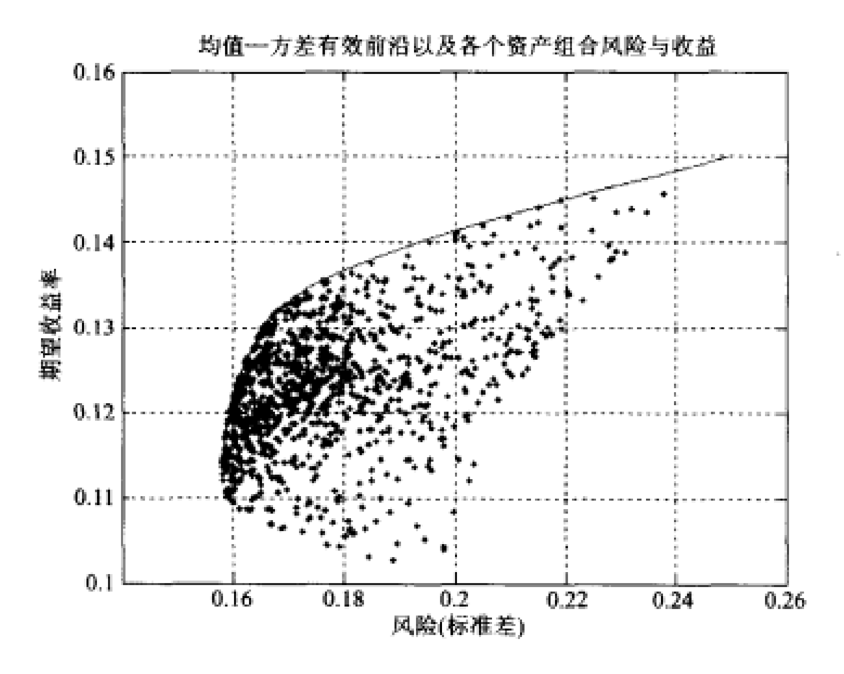


图4.5　资产组合有效前沿及各个资产组合的风险与收益

4.4.4　考虑无风险资产及存在借贷情况下的资产配置

资产组合有效前沿上的点很多，如何选择一个有效点呢？投资者需要根据自己的效用函数权衡风险与回报，在MATLAB中投资者效用函数如下：

$$U = E(r) - 0.5 \times A \times \sigma^2$$

式中：$E(r)$是未来回报，A是投资者风险厌恶系数，一般在2～4之间，σ是资产标准差。

投资者决策就是效用函数最大化，完成对资产配置。在MATLAB中考虑无风险资产时的资产配置函数是portalloc，其功能是根据风险-收益最优原则配置每项资产，其中包括无风险资产。

调用方式

```
[RiskyRisk, RiskyReturn, RiskyWts, RiskyFraction, OverallRisk, OverallReturn] =
portalloc(PortRisk, PortReturn, PortWts, RisklessRate, BorrowRate, RiskAversion)
```

输入参数

PortRisk	有效前沿上每个资产的标准差
PortReturn	有效前沿上每项资产的回报
PortWts	有效前沿上的每项资产的权重
RisklessRate	无风险利率
BorrowRate	(Optional)借款利率，默认为没有借贷

RiskAversion　(Optional)投资者的风险厌恶系数，大多数投资者风险厌恶系数在2～4之间，通常选择3

输出参数

RiskyRisk	风险资产部分标准差
RiskyReturn	风险资产部分的回报
RiskyWts	风险资产的权重
RiskyFraction	总资产中风险资产的回报
OverallRisk	总资产的标准差
OverallReturn	总资产的回报

如果资产组合中没有借贷关系，全部是风险资产，这时 RiskyFraction＝1，RiskyRisk＝OverallRisk，RiskyReturn＝OverallReturn。如果存在借贷关系的话，这时 RiskyFraction＞1，RiskyFraction＝1.5 意味着 50 %的资金是借来的。如果 RiskyFraction＝0.8，意味着风险资产的投资比率为 80 %，还有 20 %投资于无风险资产。

【例 4－10】 已知一个组合中含有 3 种资产，各资产预期回报与协方差矩阵如表 4.5 所列。

表 4.5　资产预期回报与协方差矩阵

项　目		资产 A	资产 B	资产 C
预期回报		0.1	0.2	0.15
协方差	资产 A	0.005	－0.010	0.004
	资产 B	－0.01	0.04	－0.002
	资产 C	0.004	－0.002	0.023

无风险利率为 0.08，借贷利率为 0.12，投资者风险厌恶系数为 3，要求考虑无风险资产和借贷情况下最优资产配置。

```
>> ExpReturn = [0.1 0.2 0.15];
>> ExpCovariance = [0.005    - 0.010    0.004
                   - 0.010     0.040   - 0.002
                     0.004   - 0.002    0.023];
>> [PortRisk, PortReturn, PortWts] = portopt(ExpReturn,ExpCovariance); % 由于没有输入位于有效前沿上的点的数目,MATLAB默认有效前沿上选取 10 个点,每个点代表一种组合,每个组合标准差保存在PortRisk中,收益率保存在PortReturn中,组合中各资产权重保存在PortWts中。
```

下面调用 portalloc 函数求出考虑无风险资产，以及允许借贷时的资产配置。

```
>> RisklessRate    = 0.08;
>> BorrowRate      = 0.12;
>> RiskAversion    = 3;
>> [RiskyRisk, RiskyReturn, RiskyWts, RiskyFraction, ...,
OverallRisk, OverallReturn] = portalloc(PortRisk, PortReturn,...,
PortWts, RisklessRate, BorrowRate, RiskAversion)
RiskyRisk =
            0.1283
RiskyReturn =
            0.1788
RiskyWts =
           0.0265      0.6023      0.3712
RiskyFraction =
              1.1898
OverallRisk =
             0.1527
OverallReturn =
              0.1899
```

结果表明最优组合中风险资产组合的标准差为0.128 3，收益率为0.178 8，风险资产组合中每项资产权重分别为0.026 5，0.602 3，0.371 2。总资产中风险资产配置权重为1.189 8，总资产回报为0.189 9，总资产标准差为0.152 7。

如果选取有效前沿上的20个点，得到结果如下：

```
>> [PortRisk, PortReturn, PortWts] = portopt(ExpReturn,ExpCovariance,20);
>> [RiskyRisk, RiskyReturn, RiskyWts, RiskyFraction,OverallRisk, OverallReturn]
= portalloc(PortRisk, PortReturn, PortWts, RisklessRate, BorrowRate, RiskAversion)
RiskyRisk =
            0.1288
RiskyReturn =
            0.1791
RiskyWts =
         0.0057      0.5879      0.4064
RiskyFraction =
              1.1869
```

```
OverallRisk =
              0.1529
OverallReturn =
               0.1902
```

从结果中可以知道，最优组合中风险资产部分的标准差为 0.128 3，风险资产组合收益率为 0.179 1，风险资产中每项资产权重分别为 0.005 7，0.587 9，0.406 4，总资产中风险资产配置权重为 1.186 9，总资产回报为 0.190 2，总资产标准差为 0.1529 7。除了第一项资产配置进一步减小，其他差别并不大。

4.4.5 线性规划求解资产组合问题

线性规划研究的是目标函数和约束条件均为线性的最优化问题，线性规划标准形式如下：

$$\min_{Ax=b, x\geqslant 0} \boldsymbol{Cx}$$

式中：$\boldsymbol{C}$ 是目标函数矩阵，$\boldsymbol{A}$ 是约束条件矩阵。

标准形式线性问题简称 LP(Linear Programming)问题，在 MATLAB 中用 lp 函数求解线性规划问题。

在 MATLAB 中的线性规划形式如下：

$$\begin{aligned} \min \quad & \boldsymbol{fx} \\ \text{s.t} \quad & \boldsymbol{Ax} \leqslant \boldsymbol{b} \\ & \boldsymbol{Aeq} = \boldsymbol{beq} \\ & \text{lb} \leqslant \boldsymbol{x} \leqslant \text{ub} \end{aligned}$$

式中：$\boldsymbol{f}$、$\boldsymbol{x}$、$\boldsymbol{b}$、$\boldsymbol{Aeq}$、$\boldsymbol{beq}$ 是向量，$\boldsymbol{A}$ 的形式是矩阵。

调用方式

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq,lb,ub)
```

其中，f、A、b 分别为标准线性规划模型中参数。参数 ub、lb 分别为每个变量 x 上界与下界。

【例 4-11】 某资产组合中有 3 种资产，各资产收益率分别为 0.2，0.1，0.15。要求资产 1 与资产 3 权重之和小于资产 2 权重，且没有卖空。求解使得上述收益率最大的投资组合。

首先确定目标函数为　$\max \quad 0.2x_1+0.1x_2+0.15x_3$

资产约束条件可写为

$x_1+x_3\leqslant x_2$，$x_1+x_2+x_3=1$ 且 $0\leqslant x_1$、$x_2\leqslant 1$，$0.1\leqslant x_3\leqslant 1$

在 MATLAB 中执行如下命令：

```
>> f=[-0.2 -0.1 -0.15];          % 目标函数的向量
>> a=[1 -1 1];
>> b=0;
>> aeq=[1 1 1];
>> beq=1;
>> lb=[0 0 0.1];
>> ub=[1 1 1];
>> x=linprog(f,a ,b,aeq,beq,lb,ub)
Optimization terminated.
x =
    0.4000
    0.5000
    0.1000
```

最后得出资产1、资产2、资产3权重分别为0.4、0.5、0.1。

4.4.6　二次规划求解资产组合问题

下面考虑二次规划求解资产组合。二次规划问题简称QP(Quadratic Programming),其标准形式如下:

$$\min \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{q}^{\mathrm{T}}\boldsymbol{x}$$

$$\text{s.t}\quad \boldsymbol{A}\boldsymbol{x}\leqslant \boldsymbol{b}$$

$$\boldsymbol{Aeq}*\boldsymbol{x}=\boldsymbol{beq}$$

$$\mathrm{lb}\leqslant \boldsymbol{x}\leqslant \mathrm{ub} \tag{4.5}$$

式中:$\boldsymbol{x}$为权重向量,$\boldsymbol{H}$为对称矩阵。约束条件分成两个部分:前一部分约束条件是不等式;后一部分约束条件是等式。在资产组合问题中,$\boldsymbol{H}$多为协方差阵。

在MATLAB中求解此类问题的函数是quadprog。

调用方式

```
x = quadprog(H,q,A,b)
x = quadprog(H,q,A,b,Aeq,beq)
x = quadprog(H,q,A,b,Aeq,beq,lb,ub)
输入参数 H,f,A 同式(4.5)
```

【例4-12】 资产组合中有5种资产,各资产收益率和协方差矩阵如表4.6所列。

表 4.6 各资产收益率与协方差矩阵

项目		资产 1	资产 2	资产 3	资产 4	资产 5
协方差矩阵	资产 1	0.2	0.05	−0.01	0.03	0.05
	资产 2	0.05	0.3	0.015	0.01	0.03
	资产 3	−0.01	0.015	0.1	0.02	0.01
	资产 4	0.03	0.01	0.02	0.1	0.015
	资产 5	0.05	0.03	0.01	0.015	0.15
预期收益率		0.2	0.14	0.12	0.05	0.07

要求寻找最优资产组合，使得资产组合收益率为 0.1，且该组合为方差最小资产组合。

```
>> H=[0.2,0.05,-0.01,0.03,0.05;0.05,0.3,0.015,0.01,0.03;-0.01,0.015,0.1,0.02,0.01;…
      0.03,0.01,0.02,0.1,0.015;0.05,0.03,0.01,0.015,0.15];      %各资产协方差矩阵
>> q=[0 0 0 0 0];
>> aeq=[1 1 1 1 1;0.2 0.14 0.12 0.05 0.07];                      %等式约束条件
>> beq=[1;0.1];
>> lb=[0 0 0 0 0];ub=[1 1 1 1 1];                                %权重的上下界
>> quadprog(H,q,[],[],aeq,beq,lb,ub)
ans =
    0.1067
    0.0743
    0.3379
    0.2984
    0.1826
```

上述结果表明，最优资产配置是资产 1、资产 2、资产 3、资产 4 及资产 5 所占比率分别为 10.67 %，7.43 %，33.79 %，29.84 %，18.26 %。

4.5 非线性规划求解资产组合问题

4.5.1 非线性规划基本原理

当资产组合问题的约束条件是非线性时，这时就要用到非线性规划。一般的非线性规划形式如下：

$$\min F(x) \tag{4.6}$$

$$\text{s.t}\quad \boldsymbol{Ax} \leqslant \boldsymbol{b}$$

$$\boldsymbol{Aeq} * \boldsymbol{X} = \boldsymbol{Beq}$$
$$G(\boldsymbol{x}) \leqslant 0$$
$$\mathrm{Ceq}(\boldsymbol{x}) = 0$$
$$\mathrm{VLB} \leqslant \boldsymbol{x} \leqslant \mathrm{VUB}$$

式中：$G(\boldsymbol{x})$和 $\mathrm{Ceq}(\boldsymbol{x})$都是非线性函数，$\boldsymbol{A}$ 为矩阵，$\boldsymbol{Aeq}$，$\boldsymbol{Beq}$ 分别为向量，VLB，VUB 分别为 $\boldsymbol{x}$ 的下界与上界。

4.5.2　非线性规划函数调用

调用方式

```
x = fmincon('fun',X0,A,b)
x = fmincon('fun',X0,A,b,Aeq,beq)
x = fmincon('fun',X0,A,b, Aeq,beq,VLB,VUB)
x = fmincon('fun',X0,A,b, Aeq,beq,VLB,VUB,'nonlon')
[x,fval,exitflag,output] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,@mycon)
```

输入参数同式(4.6)。

nonlon	非线性约束函数名，一般用函数 function [G Ceq] = nonlon(X)形式

输出参数

x	最优解
fval	目标函数 $F(x)$最小值
exitflag	判断收敛性，exitflag = 1 表示优化结果收敛，exitflag = 0 表示超过了最大迭代次数，exitflag<0 表示优化不收敛
output	output 为结构变量，有 3 个分量，其中 iterations 是优化过程中的迭代次数，funcCount 是代人函数值的次数，algorithm 是优化所采用的算法
options	该函数是一个结构，里面有控制优化过程的各种参数，参考 optimset()命令来设置，一般情况下不必改动它，使用默认设置就可以了

需要注意的是 fmincon 函数是局部最优，x 依赖于初始值 x_0，不同的 x_0 会有不同的最优解。optimset 函数的调用方式如下：

调用方式

```
options = optimset('param1',value1,'param2',value2,...)
```

输入参数及其内容如表 4.7 所列。

表 4.7 最优化控制参数

参数名称	内　容	说　明
Display	'off'	off 不在屏幕显示输出结果
	'iter'	iter 显示每次迭代结果
	'final'	final 显示最后一次迭代结果
	'notify'	notify 显示不收敛信息
FunValCheck	'off'	off 不提示目标函数异常值
	'on'	on 显示目标函数异常值
MaxFunEvals	正数	目标函数的上限
MaxIterpositive	正整数	最大迭代次数
OutputFcn	函数	停止迭代的控制函数
PlotFcns	函数	控制函数绘图
TolFun	实数	到达该函数值时停止迭代
TolX	实数	x 到达该值时停止迭代

例如 options = optimset('Display','iter','TolFun',1e−8)表示显示每次迭代结果，相邻两个函数值迭代误差小于 1e−8 时停止迭代过程。如果需要增加或者修改 options 的内容，规定迭代时当相邻两个 x 的距离小于 1e−4 时停止迭代，在 Command 窗口直接输入命令，optnew = optimset(options,'TolX',1e−4)。

非线性规划求解资产组合的步骤如下所述。

第一步：建立目标函数 $F(x)$。

第二步：建立约束条件。若约束条件中有非线性约束 $G(x)$或 $\mathrm{Ceq}(x)=0$，则建 M 文件 nonlcon.m 定义函数 $G(x)$与 $\mathrm{Ceq}(x)$。

第三步：建立主程序，根据要求选择 fmincon 函数的调用方式。

【例 4-13】 已知 $f(x)=-x_1x_2x_3$，约束条件 $0\leqslant x_1+2x_2+2x_3\leqslant 72$，初始值 $x=[10;10;10]$，求 $\min f(x)$。

首先建立目标函数。

```
function f = myfun(x)
f = -x(1) * x(2) * x(3);
```

改写约束条件。

$$-x_1-2x_2-2x_3\leqslant 0$$
$$x_1+2x_2+2x_3\leqslant 72$$

不等号的约束矩阵如下：

$$A=\begin{pmatrix}-1 & -2 & -2\\ 1 & 2 & 2\end{pmatrix} \qquad b=\begin{pmatrix}0\\72\end{pmatrix}$$

下面是求解步骤。

```
>> A=[-1 -2 -2;1 2 2];b=[0;72];
>> x0 = [10; 10; 10];
>> [x,fval] = fmincon(@myfun,x0,A,b)
x =
   24.0000
   12.0000
   12.0000
fval =
     -3.4560e+003
```

由结果可以知道,最优点取在(24,12,12),相应的最小值为−3 456。

【例 4-14】 资产组合中有 3 种资产,各资产收益率和协方差矩阵如表 4.8 所列。

表 4.8 各资产收益率与协方差矩阵

项目		资产 1	资产 2	资产 3
协方差	资产 1	0.02	0.05	0.01
	资产 2	0.05	0.03	0.015
	资产 3	0.01	0.015	0.01
预期收益率		0.20	0.15	0.14

投资者的目标函数

$$F=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}x-\boldsymbol{q}^{\mathrm{T}}\boldsymbol{x}$$

式中:$\boldsymbol{x}$ 是各资产权重,$\boldsymbol{H}$ 是协方差矩阵,$\boldsymbol{q}$ 是系数矩阵。

要求每个资产权重大于等于 0,没有透支,夏普比率大于 1,资产组合的收益率为 16.2 %。

首先编制两个函数文件,分别保存为目标函数和约束函数。

目标函数文件。

```
function  y=objfun(x)
t1=0.2*x(1)+0.15*x(2)+0.14*x(3);
t2=0.02*x(1)^2+0.03*x(2)^2+0.01*x(3)^2...
 +2*0.05*x(1)*x(2)+2*0.01*x(1)*x(3)+2*0.015*x(2)*x(3);
y=0.5*t2-t1;
```

约束函数文件。

```
function  [G,ceq] = confun(x)
t1 = 0.2 * x(1) + 0.15 * x(2) + 0.14 * x(3);
t2 = 0.02 * x(1)^2 + 0.03 * x(2)^2 + 0.01 * x(3)^2...
+ 2 * 0.05 * x(1) * x(2) + 2 * 0.01 * x(1) * x(3) + 2 * 0.015 * x(2) * x(3);
t2 = sqrt(t2);
G = t1/t2 - 1;
ceq = [];
```

下面求解最小值。

```
>> Aeq = [1 1 1;0.2 0.15 0.14];
>> Beq = [1;0.162];
>> VLB = [0;0;0];
>> VUB = [1;1;1];
>> x0 = [0.3 0.4 0.3];
```

下面是求解过程。

```
>> [x,fval] = fmincon(@objfun,x0,[],[],Aeq,Beq,VLB,VUB,@confun)
Warning: Large - scale (trust region) method does not currently solve this type of problem,
using medium - scale (line search) instead.
> In fmincon at 317
Optimization terminated: first - order optimality measure less
than options.TolFun and maximum constraint violation is less
than options.TolCon.
Active inequalities (to within options.TolCon = 1e - 006):
  lower      upper      ineqlin    ineqnonlin
    2
x =
    0.3667
   - 0.0000
    0.6333
fval =
   - 0.1563
```

从结果可以看出，最优组合中不含有第二项资产，第一项资产的比率为36.675 %，第三项资产的比率为63.33 %。

4.6 资产定价理论

4.6.1 证券市场线

证券市场线SML(Security Market Line)的形式如下:

$$E(r_i) = r_f + \beta[E(r_m) - r_f]$$

证券市场线是证券 i 收益率与 β_i 关系的表达式。β 的特征如下:

- 由于无风险资产与有效组合的协方差一定为零,则任何无风险资产的 β 值也一定为零;同时任何 β 值为零的资产的期望回报率也一定为零。
- 如果某种证券的协方差与有效组合的方差相等,β 值为1,则该资产的期望回报率一定等于市场有效组合的期望回报率,即这种风险资产可以获得有效组合的平均回报率。
- β 值高时,投资于该证券所获得的预期收益率就越高;β 值低时,投资于该证券所获得的预期收益率就越低。

4.6.2 CAPM模型

资本资产定价模型CAPM(Capital Asset Pricing Model)是由美国学者夏普(William Sharpe)、林特尔(John Lintner)、特里诺(Jack Treynor)和莫辛(Jan Mossin)等人在资产组合理论的基础上发展起来的,是现代金融市场价格理论的支柱,广泛应用于投资决策和公司理财领域。该模型假设非系统性风险可通过多元化投资分散掉,只有系统性风险发挥作用。

CAPM模型是对风险和收益如何定价和度量的均衡理论,其根本作用在于确认期望收益和风险之间的关系,揭示市场是否存在非正常收益。一个资产的预期回报率与衡量该资产风险的一个尺度——贝塔(Beta)值相联系。

1. CAPM模型的理论假设

它的核心假设是将证券市场中所有投资人视为初始偏好都相同的个人,并且资本资产定价模型是在Markowitz均值-方差模型的基础上发展而来,它还继承了证券组合理论的假设。具体来说包括以下几点:证券市场是有效的,即信息完全对称;存在无风险证券,投资者可以自由地按无风险利率借入或贷出资本;投资总风险可以用方差或标准差表示,系统风险可用 β 系数表示。所有的投资者都是理性的,他们均依据Markowitz证券组合模型进行均值方差分析,作出投资决策;证券交易不征税,也没有交易成本,证券市场是无摩擦的,而现实中往往根据收入的来源(利息、股息和收入等)和金额按政府税率缴税。证券交易要依据交易量的大小和客户的资信交纳手续费、佣金等费用;除了上述这些明确的假设之外,还有如下隐含性假设:每种证券的收益率分布均服从正态分布;交易成本可以忽略不计;每项资产都是无限可分的,

这意味着在投资组合中，投资者可持有某种证券的任何一部分。

2. CAPM模型方程

在上述假设条件下，可以推导出CAPM模型的具体形式：

$$E(r_i) - r_f = \beta_i[E(r_m) - r_f]$$

$$\beta_i = \mathrm{cov}(r_i, r_m)/\mathrm{Var}(r_m) = \sigma_{im}/\sigma_m^2$$

式中：$E(r_i)$为证券i的期望收益，$E(r_m)$为市场组合的期望收益，r_f为无风险资产的收益，$\sigma_{im} = \mathrm{cov}(r_i, r_m)$为证券$i$收益率和市场组合收益率的协方差，$\sigma_m^2 = \mathrm{Var}(r_m)$为市场组合收益率的方差。

CAPM模型与证券市场线的区别在于CAPM考虑$E(r_m)$与风险溢价之间的关系，证券市场线是研究$E(r_i)$与β_i之间的关系。

3. CAPM模型的理论意义

资本资产定价理论认为，一项投资所要求的必要报酬率取决于以下3个因素：

① 无风险利率，即将国债投资（或银行存款）视为无风险投资。

② 市场平均收益率，即整个市场的收益率。

③ 投资组合的系统风险系数即β系数，是某一投资组合的风险程度与市场证券组合的风险程度之比。

CAPM模型说明了单个证券投资组合的期望收益率与相对风险程度间的关系，即任何资产的期望收益率一定等于无风险利率加上一个风险调整，后者相对整个市场组合的风险程度越高，需要得到的额外补偿也就越高。这也是资产定价模型（CAPM）的主要结果。

4. CAPM理论的主要作用

CAPM理论是现代金融理论的核心内容，它的作用主要在于：通过预测证券的期望收益率和标准差的定量关系来考虑已经上市的不同证券价格的"合理性"；可以帮助确定准备上市证券的价格；能够估计各种宏观经济变化对证券价格的影响。

由于CAPM从理论上说明在有效率资产组合中，β描述了任意一项资产的系统风险（非系统风险已经在分化中相互抵消掉了），任何其他因素所描述的风险尽为β所包容。并且模型本身要求存在一系列严格的假设条件，所以CAPM模型存在理论上的抽象和对现实经济的简化。虽然我国股市和CAPM的假设条件有相当的差距，但没有必要等到市场发展到某种程度再来研究CAPM在我国的实际应用问题，相反，充分利用CAPM较强的逻辑性、实用性，通过对市场的实证分析和理论研究，有利于发现问题，推动我国股市的发展。

5. CAPM模型估计

标准的CAPM模型是线性模型，对于n种资产，m为每个样本的观察值，第k项资产的第

t 时刻观察的收益率为 $R_{k,t}$，市场的回报为 M_t，无风险利率 C_t，CAPM 模型如下：

$$R_{k,t} = a_k + C_t + \beta_k(M_t - C_t) + \epsilon_{k,t}$$

式中：$k=1,2,\cdots,n$；$t=1,2,\cdots,m$；α_k、β_k 分别为资产 k 的 Alpha 与 Beta；ϵ 为白噪声。要求 $E(e_{k,.})=0$，$E(\epsilon_i,\epsilon_j)=\sigma_{i,j}$ 为残差。

实际上回归的形式如下：

$$R_{k,t} - C_t = a_k + \beta_k(M_t - C_t) + \epsilon_{k,t}$$

6. 估计 CAPM 模型的 Beta 和方差

【例 4-15】 利用 MATLAB 自带的数据估计 CAPM 模型的参数，数据文件名称是 CAPMuniverse，其中有 3 个变量。变量 Assets 是 14 个品种名称，前面 12 个是股票，后面 2 个分别代表市场与货币市场；Data 是股票自 2000 年 1 月 1 日至 2005 年 11 月 7 日的日收益率，前面 12 列为股票数据，其中第二只和第六只股票数据不全，其中一个是因为发行新股(IPO)；Dates 是日期，后面两个分别为股票市场与货币市场收益率。下面首先去掉缺失数据的股票然后估计 CAPM 模型中的参数。

首先调用数据

```
>> load CAPMuniverse
% 变量 Assets 是股票的名称，Data 是股票的价格，Dates 是序数型日期
>> Data(:,[2,6]) = []                % 去掉第二只股票数据
>> Assets(:,[2,6]) = []              % 去掉第六只股票数据
```

下面是程序。

```
[NumSamples, NumSeries] = size(Data);
NumAssets = NumSeries - 2;              % 确定股票的个数，不含市场、货币市场收益率
% 确定起始日和结束日
StartDate = Dates(1);
EndDate = Dates(end);
% 确定输出表格头
fprintf(1,'Separate regressions with ');
fprintf(1,'daily total return data from %s to %s \n', datestr(StartDate,1),datestr(EndDate,1));
fprintf(1,'  %4s % -20s % -20s % -20s\n','','Alpha','Beta','Sigma');
fprintf(1,'  ---- ------------------------ ');
fprintf(1,'-----------------------------------\n');
% 对每个资产进行估计
for i = 1:NumAssets
  TestData = zeros(NumSamples,1);
  TestDesign = zeros(NumSamples,2);
```

```
  TestData(:) = Data(:,i) - Data(:,12);                % 股票的超额收益率
  TestDesign(:,1) = 1.0;                               % 表示模型中含有常数项
  TestDesign(:,2) = Data(:,11) - Data(:,12);           % 市场的超额收益率
% 带有缺失数据的多元回归模型估计 CAPM,Param 是系数,Covar 是残差
  [Param, Covar] = mvnrmle(TestData, TestDesign);
% 根据残差估计多元正态分布的协方差矩阵
  [StdParam, StdCovar] = mvnrstd(TestData, TestDesign, Covar);
% 确定输出结果变量 Assets 是股票的名称,Data 是股票的价格,Dates 是序数型日期
  Alpha = Param(1);
  Beta = Param(2);
  Sigma = sqrt(Covar);
  StdAlpha = StdParam(1);
  StdBeta = StdParam(2);
  StdSigma = sqrt(StdCovar);
% 确定显示格式
  fprintf('  %4s %9.4f (%8.4f) %9.4f (%8.4f) %9.4f (%8.4f)\n', ...
     Assets{i},Alpha(1),abs(Alpha(1)/StdAlpha(1)), ...
     Beta(1),abs(Beta(1)/StdBeta(1)),Sigma(1),StdSigma(1));
end
```

结果显示如下：

```
Separate regressions with daily total return data from 03-Jan-2000 to 07-Nov-2005...
             Alpha                 Beta                  Sigma
-----------------------------------------------------------------------
AAPL         0.0012 (1.3882)       1.2294 (17.1839)      0.0322(0.0062)
CSCO         -0.0002(0.2878)       1.5653(23.6085)       0.0298(0.0057)
DELL         -0.0000(0.0368)       1.2594(22.2164)       0.0255(0.0049)
EBAY         0.0014(1.4326)        1.3441(16.0732)       0.0376(0.0072)
HPQ          0.0001(0.1747)        1.3745(24.2390)       0.0255(0.0049)
IBM          -0.0000(0.0312)       1.0807(28.7576)       0.0169(0.0032)
INTC         0.0001(0.1608)        1.6002(27.3684)       0.0263(0.0050)
MSFT         -0.0002(0.4871)       1.1765(27.4554)       0.0193(0.0037)
ORCL         0.0000(0.0389)        1.5010(21.1855)       0.0319(0.0061)
YHOO         0.0001(0.1282)        1.6543(19.3838)       0.0384(0.0074)
```

4.6.3 计算经过风险调整的 Alpha 及回报

衡量投资组合的优劣不能仅仅凭借收益率指标，高收益可能是因为承担过高的风险，需要

剔除风险因素，然后再进行比较。调整的方式很多，有时还需参照无风险利率、市场收益率，无风险利率一般用短期国债，或者信用等级（AAA 级商业票据）比较高的商业票据收益率。通过比较经过风险调整的 Alpha 及回报可以衡量不同投资组合的优劣。

调用方式

```
portalpha(Asset, Benchmark)
portalpha(Asset, Benchmark, Cash)
portalpha(Asset, Benchmark, Cash, Choice)
Alpha = portalpha(Asset, Benchmark, Cash, Choice)
[Alpha, RAReturn] = portalpha(Asset, Benchmark, Cash, Choice)
```

输入参数

Asset	资产的回报
Benchmark	基准市场的回报
Cash	(Optional) 无风险利率，默认值为 0
Choice	(Optional) ALPHA 的计算方法，默认值为'xs'
	'xs'没有经过风险调整超额回报(no risk adjustment)
	'sml'证券市场线(SML)
	'capm'詹森法计算的 ALPHA(Jensen's Alpha)
	'mm'穆迪尼安尼-米勒法(Modigliani & Miler)
	'gh1'Graham - Harvey 1 法
	'gh2'Graham - Harvey 2 法
	'all'分别计算上面所有的回报

如果选择'xs'，那么 Alpha 和收益的计算公式如下：

$$\alpha = R - R_M$$

$$\hat{R} = R$$

式中：α 为组合的溢价，$\hat{R}$ 为没有经过风险调整的收益率，R_M 为市场收益率。

如果选择'sml'，那么 Alpha 和经过风险调整的收益的计算公式如下：

$$\alpha = R - R_f - (R - R_f)\frac{R_M - R_f}{\sigma_M - \sigma_f}$$

$$\hat{R} = R - \alpha$$

式中：α 为 Alpha，$\hat{R}$ 为经过风险调整的收益率，R 为组合收益率，R_f 为无风险收益率，R_M 为市场收益率，σ_M 为市场收益率的标准差，σ_f 为无风险收益率的标准差。

如果选择'capm'，那么 Alpha 和经过风险调整的收益的计算公式如下：

$$\alpha = R - R_f - \frac{\text{cov}(R, R_M)}{\sigma_R \sigma_M}(R_M - R_f)$$

$$R = R - \alpha$$

式中：σ_R 为组合收益率的标准差，$\text{cov}(R,R_M)$为组合收益率与市场收益率之间的协方差。

如果选择'mm'，那么 Alpha 和经过风险调整的收益的计算公式如下：

$$\alpha = \frac{\sigma_M}{\sigma_R}(R - R_f) - (R_M - R_f)$$

$$\hat{R} = \alpha + R_f$$

如果选择'gh1'，那么 Alpha 和经过风险调整的收益的计算公式如下：

$$\alpha = (R - R_f) - \text{Factor} \times (R_M - R_f)$$

$$\hat{R} = R - \alpha$$

式中：

$$\text{Factor} = \frac{L_1 + \sqrt{|\text{disc}|}}{L_1 + L_2}$$

$$\text{disc} = L_1^2 + (L_1 + L_2)(\sigma_M^2 - \sigma_f^2)$$

$$L_1 = \sigma_f^2 - \text{cov}(R_M, R_f)$$

$$L_2 = \sigma_M^2 - \text{cov}(R_M, R_f)$$

如果选择'gh2'，那么 Alpha 的计算公式和经过风险调整的收益如下：

$$\alpha = \text{Factor} \times (R - R_f) - (R_M - R_f)$$

$$\hat{R} = \alpha + R_M$$

式中：

$$\text{Factor} = \frac{L_1 + \sqrt{|\text{disc}|}}{L_1 + L_2}$$

$$L_1 = \sigma_f^2 - \text{cov}(R, R_f)$$

$$L_2 = \sigma^2 - \text{cov}(R, R_f)$$

输出参数

Alpha	资产的 Alpha
RAReturn	经过风险调整的收益

下面计算不同方法下的 Alpha 和回报。

```
>> load FundMarketCash
% 将价格序列转化为收益率序列
>> Returns = tick2ret(TestData);
% 计算没有经过风险调整的 Alpha 和回报
>> [Alpha, RAReturn] = portalpha(Returns(:,1),Returns(:,2),Returns(:,3))
Alpha =
```

```
    8.0632e-004
RAReturn =
    0.0038
% 计算资本市场线下的 Alpha 和回报
>> [Alpha, RAReturn] = portalpha(Returns(:,1),Returns(:,2),Returns(:,3),'sml')
Alpha =
        0.0013
RAReturn =
          0.0025
% 根据 CAPM 模型计算 Alpha
>> [Alpha, RAReturn] = portalpha(Returns(:,1),Returns(:,2),Returns(:,3),'capm')
Alpha =
        0.0013
RAReturn =
          0.0024
```

4.7 蒙特卡洛模拟多资产组合

在 MATLAB 中可以模拟多资产的价格运动。如果 n 个资产的价格满足下列方程：

$$\frac{dS}{S} = \mu dt + \sigma \epsilon dz$$

式中：$S=(S_1,S_2,S_3,\cdots,S_n)$表示资产价格，$\sigma$ 表示标准差，$\epsilon\sim N(0,\sum)$服从 n 元正态分布，dt 表示时间间隔。

调用方式

```
RetSeries = portsim(ExpReturn, ExpCovariance, NumObs, RetIntervals, NumSim,Method)
```

输入参数

ExpReturn	各资产期望收益
ExpCovariance	各资产的协方差
NumObs	每条路径上观察值的个数，如果是[]，由 RetIntervals 确定
RetIntervals	(Optional)模拟的时间间隔，默认值为 1，即是 Δt 的长度，如果每个资产的时间间隔不同时，RetIntervals 是一个向量
NumSim	(Optional)模拟运动轨迹的个数，默认值为 1
Method	(Optional)模拟的方式，当 Method = 'Exact'时，模拟样本的均值和协方差等于输入值，当样本数目多时处理速度较慢；当 Method = 'Expect'时，样本的均值和协方差渐近无偏，速度较快

输出参数

RetSeries 模拟的结果

【例 4-16】 已知各资产的期望收益的收益、标准差及相关系数矩阵如表 4.9 所列。

表 4.9 资产的收益、标准差及相关系数表

项目		资产 A	资产 B	资产 C	资产 D	资产 E
期望收益		0.000 246	0.000 189	0.000 273	0.000 141	0.000 311
标准差		0.009 509	0.014 259	0.015 227	0.011 062	0.010 877
相关系数	资产 A	1	0.440 3	0.473 5	0.433 4	0.685 5
	资产 B	0.440 3	1	0.759 7	0.780 9	0.434 3
	资产 C	0.473 5	0.759 7	1	0.697 8	0.492 6
	资产 D	0.433 4	0.780 9	0.697 8	1	0.428 9
	资产 E	0.685 5	0.434 3	0.492 6	0.428 9	1

下面模拟其组合的运动轨迹。

第一步:建立期望、标准差及相关系数矩阵。

```
>> ExpReturn    = [0.0246   0.0189   0.0273   0.0141   0.0311]/100;
>> Sigmas       = [0.9509   1.4259   1.5227   1.1062   1.0877]/100;
>> Correlations = [1.0000   0.4403   0.4735   0.4334   0.6855
                   0.4403   1.0000   0.7597   0.7809   0.4343
                   0.4735   0.7597   1.0000   0.6978   0.4926
                   0.4334   0.7809   0.6978   1.0000   0.4289
                   0.6855   0.4343   0.4926   0.4289   1.0000];
```

第二步:根据标准差、相关系数计算协方差。

```
>> ExpCovariance = corr2cov(Sigmas, Correlations)
ExpCovariance =
    1.0e-003 *
    0.0904    0.0597    0.0686    0.0456    0.0709
    0.0597    0.2033    0.1649    0.1232    0.0674
    0.0686    0.1649    0.2319    0.1175    0.0816
    0.0456    0.1232    0.1175    0.1224    0.0516
    0.0709    0.0674    0.0816    0.0516    0.1183
```

第三步:输入模拟的目标参数。

```
>> StartPrice      = 100;        % 各资产的起始价格为100,如果不是100,就要用向量
```

```
>> NumObs        = 504;    % 每个资产的路径模拟 504 个观察值
>> NumSim        = 2;      % 模拟 2 条组合轨迹
>> RetIntervals  = 1;      % 相邻两点的时间间隔为 1,总的时间长度为 1×504
>> NumAssets     = 5;      % 资产的个数是 5
```

第四步：开始模拟每个资产的运动轨迹。采用两种模拟方法。

```
>> randn('state',0);
>> RetExact = portsim(ExpReturn, ExpCovariance, NumObs, ...
RetIntervals, NumSim, 'Exact');                       % 精确法模拟的结果
>> randn('state',0);
>> RetExpected = portsim(ExpReturn, ExpCovariance, NumObs, ...
RetIntervals, NumSim, 'Expected');                    % 渐近法模拟结果
```

第五步：根据各资产权重计算组合的收益率。

```
>> Weights          = ones(NumAssets, 1)/NumAssets;
>> PortRetExact     = zeros(NumObs, NumSim);
>> PortRetExpected  = zeros(NumObs, NumSim);
>> for i = 1:NumSim
    PortRetExact(:,i)    = RetExact(:,:,i) * Weights;
    PortRetExpected(:,i) = RetExpected(:,:,i) * Weights;
end
```

第六步：根据收益率、初始价格计算模拟价格。

```
>> PortExact    = ret2tick(PortRetExact, repmat(StartPrice,1,NumSim));
>> PortExpected = ret2tick(PortRetExpected,repmat(StartPrice,1,NumSim));
```

第七步：绘出每种资产的价格图。

```
>> subplot(2,1,1), plot(PortExact, '-r')
>> ylabel('Portfolio Prices')
>> title('Exact Method')                 % 采用精确方法模拟的价格
>> subplot(2,1,2), plot(PortExpected, '-b')
>> ylabel('Portfolio Prices')
>> title('Expected Method')              % 采用渐近方法模拟
```

两种方法的模拟结果如图 4.6 所示。

【例 4-17】 各资产的期望收益、标准差及相关系数矩阵同上例，一年有 252 个交易日，下面模拟 4 年的情况。分别用两种方法各模拟 1 条轨迹，一种是时间间隔为 1 个交易日，另一种是以 1 年（252 个交易日）为 1 单位，模拟的时间间隔是 1 年的 1/252；从理论上看这两种方

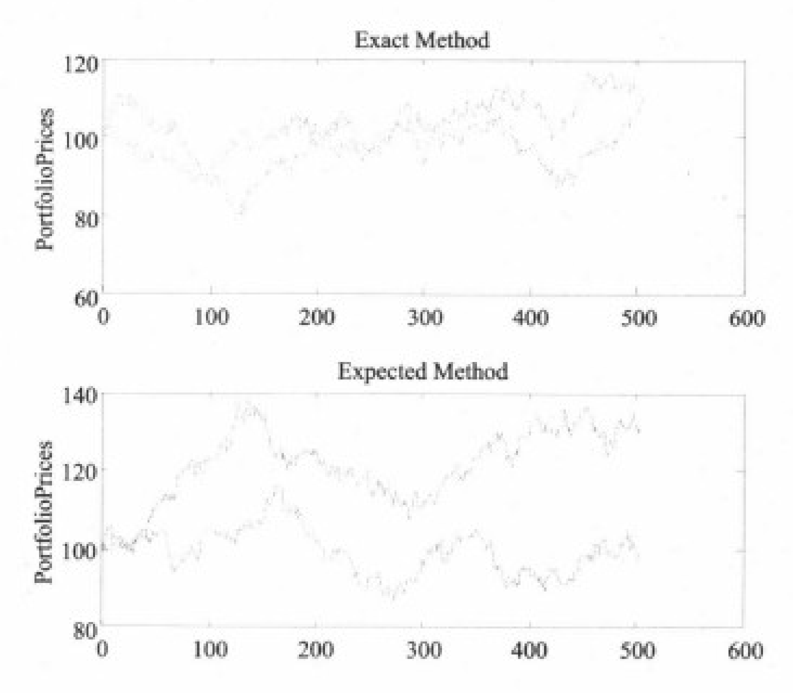


图 4.6 蒙特卡罗方法模拟多资产组合的价格轨迹

法是一回事。

前面两步同【例 14-16】。

第三步：输入模拟的目标参数。

```
>> StartPrice       = 100;
>> NumObs           = 1008;       % 模拟 4 年(1008 = 4×252)的价格
>> RetIntervals     = 1;          % 几个交易日
>> NumAssets        = 5;
>> randn('state',0);
>> % 以交易日为 1 单位。
>> RetSeries1 = portsim(ExpReturn, ExpCovariance, NumObs, ...
RetIntervals, 1, 'Expected');
>> % 以一年 252 个交易日为 1 单位。选取的时间间隔为原来的 1/252
>> randn('state',0);
>> RetSeries2 = portsim(ExpReturn * 252, ExpCovariance * 252, ...
NumObs, RetIntervals/252, 1, 'Expected');
```

第四步：根据各资产权重计算组合的收益率轨迹。

```
>> Weights = ones(NumAssets, 1)/NumAssets;
>> PortRet1 = RetSeries2 * Weights;
>> PortRet2 = RetSeries2 * Weights;
```

下面比较二者的结果。

```
>> tt = PortRet1 - PortRet2;
>> find(tt~ = 0)
ans =
     Empty matrix: 0 - by - 1
```

从上面可以看出二者结果是相同的。

【例4-18】 模拟单资产运动轨迹。资产的初始价格为20,1年期望收益为0.2,标准差为0.4,每个轨迹取样252个样本点,相邻的样本点时间间隔为1天,模拟10 000个轨道。模拟1年中每个交易日的股价。

输入模拟的目标参数。

```
>> StartPrice       = 20;
>> ExpReturn        = 0.2;
>> ExpCovariance    = 0.4^2;
>> NumObs           = 252;
>> NumSim           = 10000;
>> RetIntervals     = 1/252;
>> randn('state',10);
>> RetSeries = portsim(ExpReturn, ExpCovariance, NumObs, ...
RetIntervals, NumSim, 'Expected');
% RetSeries 为 252×1×10 000 维矩阵
>> RetSeries = squeeze(RetSeries);                    % 转化为 252×10 000 维矩阵
>> % 根据收益率计算价格
>> StockPrices = ret2tick(RetSeries, repmat(StartPrice, 1, NumSim));
```

如果资产运动轨迹方程为

$$\frac{dS}{S} = \mu dt + \sigma \epsilon dz$$

式中:S为股价,μ,σ分别为运动的期望与标准差,ϵ为标准正态分布。Hull的《期权、期货和其他衍生产品》一书中的12.2节给出的股价的期望与方差为

$$E(S_T) = S_0 e^{\mu T}$$

$$\mathrm{Var}(S_T) = S_0^2 e^{2\mu T}(e^{\sigma^2 T} - 1)$$

下面比较理论值与模拟值。

首先计算模拟价格的均值与方差。

```
>> StockPrices = ret2tick(RetSeries, repmat(StartPrice, 1, NumSim));
>> SampMean = mean(StockPrices(end,:))
SampMean =
```

```
        24.4587
>> SampVar = var(StockPrices(end,:))
SampVar =
      104.2016
```

然后计算理论的价格与方差。

```
>> ExpValue = StartPrice * exp(ExpReturn)
ExpValue =
        24.4281
>> ExpVar = StartPrice * StartPrice * exp(2 * ExpReturn) * (exp((ExpCovariance)) - 1)
ExpVar =
      103.5391
```

可以看出理论值与模拟的结果非常类似。
如果对频率进行统计，有

```
>> hist(StockPrices(end,:), 30)              % 直方图矩形方块 30 个
```

最终价格的频率直方图如图 4.7 所示。

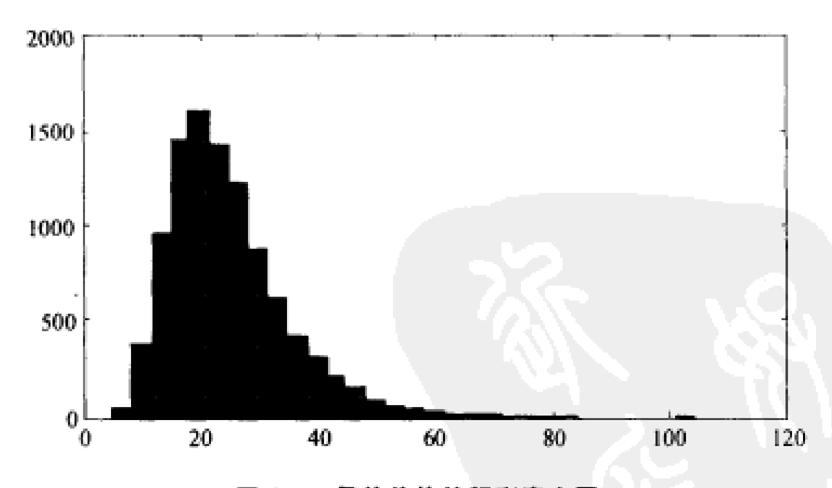


图 4.7 最终价格的频率直方图

下面画出频率的概率直方图。

```
>> [count, BinCenter] = hist(StockPrices(end,:), 30);          % 统计股价的频率
>> figure
>> bar(BinCenter, count/sum(count), 1, 'r')
>> xlabel('股票的终端价格')
>> ylabel('概率')
>> title('收益率服从对数正态分布的股价分布图')
```

模拟股价最终价格概率直方图如图4.8所示。

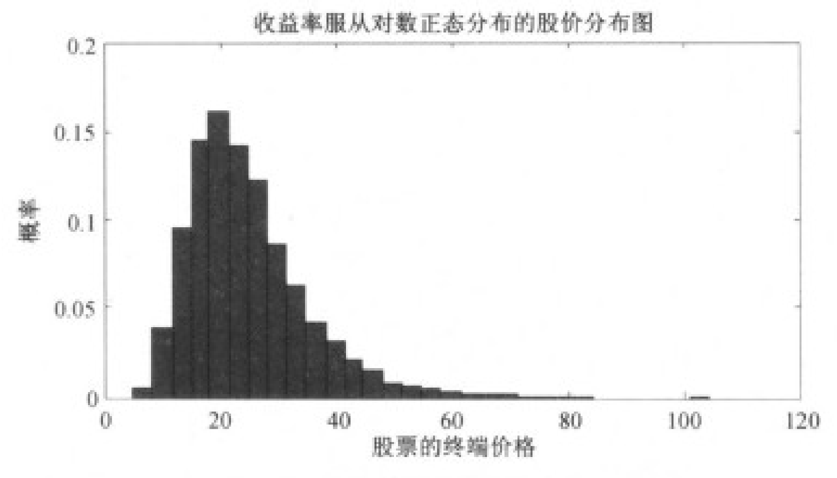


图4.8　股价最终价格概率直方图

思考题

1. 已知资产日回报率为0.002 5，标准差为0.020 8，资产现在价值为0.8亿元，求5％水平下资产的在险价值（VaR）。

2. 某资产组合中4种资产协方差矩阵如表4.10所列。

表4.10　各资产协方差、预期收益

项　目		资产1	资产2	资产3	资产4
协方差矩阵	资产1	0.2	0.05	−0.01	0.03
	资产2	0.05	0.3	0.015	0.01
	资产3	−0.01	0.015	0.1	0.02
	资产4	0.03	0.01	0.02	0.1
预期收益率		0.2	0.14	0.12	0.05

试计算相关系数，以及资产组合有效前沿，如果规定第4种资产权重小于20％，则其有效前沿如何变化。

3. 比较第2题的投资组合的收益、方差与个股收益、方差，如果无风险利率为0.02，则试计算个股与组合的夏普比率。

第 5 章　金融数据统计

本章主要介绍了统计学的基本原理和基本统计量。要求读者掌握均匀分布、正态分布随机数生成方法，学会常用的统计绘图命令，掌握回归的方法，学会运用主成分、因子分析金融问题。

5.1　随机模拟基本原理

1977 年，菲力浦·伯耶勒(Phelim Boyle)提出了模拟方法求解金融资产定价问题。其想法是假设资产价格分布是随机波动，如果知道了这个波动过程，就可以模拟不同的路径；每做完一次模拟，就产生一个最终资产价值，再进行若干次这样的过程，那么所得到的结果就是一个最终资产价值分布，从这个分布中可以得到期望的资产价格。

5.1.1　随机数生成函数

1. 均匀分布随机数生成函数

在 MATLAB 中的 unidrnd 函数可以生成 1～N 的均匀分布随机数。

调用方式

```
R = unidrnd(N)
R = unidrnd(N,m)
R = unidrnd(N,m,n)
```

输入参数

N	生成在 1～N 之间的一个随机数
m	确定输出随机矩阵 $\boldsymbol{R}$ 的行数
n	确定输出随机矩阵 $\boldsymbol{R}$ 的列数

输出参数

R	随机数矩阵

2. 生成服从连续均匀分布的随机数

如果需要生成服从连续均匀分布的随机数，则可以调用 unifrnd 函数，其调用方式如下：

调用方式

```
R = unifrnd (A,B)
R = unifrnd (A,B,m)
R = unifrnd (A,B,m,n)
```

A,B 是随机数的下界与上界。m,n 表示随机数的维数。
如生成一个 0～1 之间随机数。

```
>> unifrnd(0,1)
ans =
      0.4565
```

下面介绍两种方法生成 1～2 之间随机矩阵 $\boldsymbol{K}$,$\boldsymbol{K}$ 为 5 行 6 列矩阵。
方法 1

```
>> unifrnd(1,2,[5,6])
ans =
      1.0185    1.9218    1.9169    1.8132    1.6038    1.4451
      1.8214    1.7382    1.4103    1.0099    1.2722    1.9318
      1.4447    1.1763    1.8936    1.1389    1.1988    1.4660
      1.6154    1.4057    1.0579    1.2028    1.0153    1.4186
      1.7919    1.9355    1.3529    1.1987    1.7468    1.0000
```

方法 2

```
>> unifrnd(1,2,5,6)
ans =
      1.5252    1.6813    1.4289    1.3028    1.8600    1.8216
      1.2026    1.3795    1.3046    1.5417    1.8537    1.6449
      1.6721    1.8318    1.1897    1.1509    1.5936    1.8180
      1.8381    1.5028    1.1934    1.6979    1.4966    1.6602
      1.0196    1.7095    1.6822    1.3784    1.8998    1.3420
```

3. 生成正态分布的随机数

调用方式

```
R = normrnd(mu,sigma)
R = normrnd(mu,sigma,m)
R = normrnd(mu,sigma,m,n)
```

输入参数

mu	正态分布的均值
sigma	正态分布的方差
m	随机矩阵 ***R*** 的行数
n	随机矩阵 ***R*** 的列数

如需生成均值为0、方差为1正态分布的随机数，则可以执行下面命令。

```
>> normrnd(0,1)
ans =
      - 0.4326
```

下面用两种方法生成均值为0、方差为1的正态分布矩阵，矩阵为5行6列。

方法1

```
>> normrnd(0,1,[5 6])
ans =
    - 1.6656    1.1892    0.7258    1.0668   - 1.3362    1.2540
      0.1253  - 0.0376  - 0.5883    0.0593     0.7143  - 1.5937
      0.2877    0.3273    2.1832  - 0.0956     1.6236  - 1.4410
    - 1.1465    0.1746  - 0.1364  - 0.8323   - 0.6918    0.5711
      1.1909  - 0.1867    0.1139    0.2944     0.8580  - 0.3999
```

方法2

```
>> normrnd(0,1,5, 6)
ans =
     0.6900    1.1908    0.2573    0.2193    0.6145    0.3803
     0.8156  - 1.2025  - 1.0565  - 0.9219    0.5077  - 1.0091
     0.7119  - 0.0198    1.4151  - 2.1707    1.6924  - 0.0195
     1.2902  - 0.1567  - 0.8051  - 0.0592    0.5913  - 0.0482
     0.6686  - 1.6041    0.5287  - 1.0106  - 0.6436    0.0000
```

4. 特定分布随机数发生器

在MATLAB中有统一格式随机数发生器，函数名称为random，可生成许多服从不同分布的随机数。

调用方式

```
y = random('name',A1,A2,A3,m,n)
```

输出参数

name　　说明随机分布类型，具体如表 5.1 所列。

表 5.1　特定分布的参数表

类　别	贝　塔	二项分布	卡　方	指数分布	F 分布	伽　玛	对数正态
分布	Beta	Binomial	Chisquare	Exponential	F	Gamma	Lognormal
简写	beta	bino	chi2	exp	f	gam	logn

类　别	均匀分布	泊松分布	T 分布	正态分布	非中心 F 分布	非中心 T 分布
分布	Uniform	Poisson	T	Normal	Noncentral F	Noncentral T
简写	unif	poiss	t	norm	ncf	nct

下面用 random 函数生成 3 行 2 列的正态分布随机数矩阵，正态分布的均值为 0、方差为 1。

```
>> a = random('Normal',0,1,3,2)
a =
    - 0.4326     0.2877
    - 1.6656   - 1.1465
      0.1253     1.1909
```

5. 多元正态分布的随机数

多元正态分布的随机数可以用如下形式表示：

$$X_i \sim N(\boldsymbol{\mu}, \Sigma)$$

式中：$\boldsymbol{\mu}$ 是均值向量，Σ 是协方差矩阵。

在 MATLAB 中可使用 mvnrnd 函数生成多元正态分布函数。

调用方式

```
R = mvnrnd(mu,sigma)
R = mvnrnd(mu,sigma,cases)
```

输入参数

mu　　均值
sigma　　协方差
cases　　样本个数

下面生成一个多元正态分布的例子。

```
>> mu = [2 3];                              % 均值
>> SIGMA = [1 1.5; 1.5 3];                  % 协方差矩阵
>> r = mvnrnd(mu,SIGMA,100);                % 生成 100 个随机样本
>> plot(r(:,1),r(:,2),'+')
```

样本的散点图如图 5.1 所示。

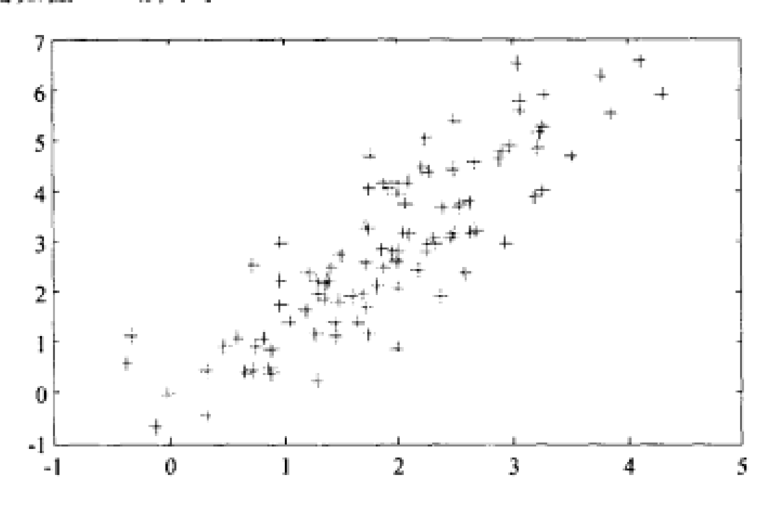


图 5.1　二元正态分布的散点图

5.1.2　多元正态分布密度函数

多元正态分布的密度函数是 mvnpdf。

调用方式

```
mvnpdf(X,mu,Sigma)
```

下面是一个例子。

```
>> mu = [1 -1];
>> Sigma = [.9 .4; .4 .3];
>> X=[ 2 1];
>> p = mvnpdf(X,mu,Sigma)
p =
    1.3828e-005
```

如果计算分布函数，则 X,Y 为二元随机正态分布，分布函数 $F(x,y)$ 的定义如下：

$$F(x,y) = P(X < x, Y < y)$$

调用方式

```
p = mvncdf(X,mu,SIGMA)
```

下面是一个例子。

```
>> mu = [1 -1];
>> Sigma = [.9 .4; .4 .3];
>> X=[2 1];
>> f = mvncdf(X,mu,Sigma)
f =
    0.8541
```

可以看出对于随机变量 X,Y,有

$$P(X < 2, Y < 1) = 0.8541$$

也即 $X<2$ 且 $Y<1$ 的概率为 0.854 1。

5.2　随机变量的数字特征

5.2.1　计算平均值

调用方式

```
M = mean(A)
mean(A,dim)
```

输入参数

A	如果 **A** 为向量,则返回值是该向量的平均值;如果 **A** 是矩阵,则返回值是每列的平均值
dim	dim = 1(默认)表示每列平均, dim = 2 表示每行平均

下面是一个例子。

```
>> a=[1 2 ;3 4]
a =
    1     2
    3     4
>> mean(a)
ans =
      2     3
>> mean(a,2)
ans =
      1.500 0
      3.500 0
```

在 MATLAB 中计算几何平均的函数为 geomean，计算调和平均数的函数是 harmmean 函数，调和平均数的计算公式为 $M=\dfrac{n}{\sum\frac{1}{x_i}}$，注意样本数据不能为 0。

5.2.2 剔除异常值后的平均值

有时观察数据中有异常大或者异常小的值，这些异常值会对平均数产生影响，需要去掉异常值。例如在体操比赛中，去掉一个最高分和一个最低分，然后计算运动员的最后得分。在 MATLAB 中也有剔除异常值后的平均数。

调用方式

```
M = trimmean(X,percent)
M = trimmean(X,percent,dim)
```

输入参数

X	样本观察值矩阵
percent	剔除比率，例如 percent = 10 表示同时剔除最大的 5 % 与最小的 5 % 观察值
dim	dim = 1(默认)表示对每列求平均，dim = 2 表示对每行求平均

```
>> x = rand(1,20);    % 生成 20 个随机数
>> trimmean(x,10)     % 剔除 5 % 的异常大值与 5 % 异常小值
ans =
      0.5145
```

剔除 10 %的异常值之后的平均数为 0.514 5。

5.2.3 计算中位数

调用方式

```
M = median(A)
M = median(A,dim)
```

输入参数

A	样本观察值矩阵
dim	dim = 1(默认)表示对每列求平均，dim = 2 表示对每行求平均

有时数据中出现 NaN，在计算中位数时需要忽略 NaN，这时需要调用 nanmedian 函数。

5.2.4 计算方差与标准差

一般说来，资产组合的风险越大，方差越大，波动性越大。方差由于其简单、直观以及良好

的统计性质使其成为风险的代名词。

在 MATLAB 中计算方差标准差的函数分别为 Var,Std。

方差调用方式

```
Var (A)
Var(A,flag)
```

标准差调用方式

```
Std(A)
Std(A,flag)
```

输入参数

A　　样本值

flag　　0(默认值)表示方差计算公式为 $\mathrm{Var}=\frac{\sum_{i=1}^{N}(x_i-\bar{x})^2}{N-1}$

　　1 表示方差计算公式为 $\mathrm{Var}=\frac{\sum_{i=1}^{N}(x_i-\bar{x})^2}{N}$

5.2.5　计算样本的百分位数

调用方式

```
Y = prctile(X,p,dim)
```

输入参数

X　　观察值

p　　计算大于 p% 值的数

dim　　同上

下面是一个例子。

```
>> x = rand(1,20);
>> prctile(x,20)
ans =
      0.3562
```

5.2.6　计算样本极差

极差就是样本极大值与极小值的差,反映样本的离散程度。

调用方式

```
[min, max] = range(a)
r = range(q)
[min, max] = range(q)
```

下面是一个例子。

```
>> x = rand(1,20);
>> range(x)
ans =
      0.6100
```

5.2.7 计算偏度与峰度

方差作为风险的度量指标并不是完整的。比如讲，两种资产收益分布的均值和方差都是相同的，但是一种资产收益是左偏的，另一种是右偏的。对于风险而言，相对于大概率小损失人们更加厌恶小概率大损失的情况，后一种情况给人们带来的痛苦远大于第一种情况。从这个意义上讲，收益分布左偏的资产的风险水平要小于右偏的资产。此时，方差作为风险的度量指标就不是完全的，还要考虑峰度、偏度等指标。

偏度和峰度是两个高阶的统计量。计算偏度的目的在于考查组合收益率水平是否是对称分布，也就是说组合产生亏损与获得盈利的概率如何；峰度是考查组合的收益率情况是否接近正态分布，如果组合的收益率存在尖峰厚尾的分布特征，则说明组合产生亏损和盈利的概率偏大，也就在一定程度上认为组合收益率出现极端性的可能性偏大，这种组合的收益率稳定性是比较差的。

偏度的计算公式为

$$\text{skewness} = \frac{E(x-\mu)^3}{\sigma^3}$$

式中：μ,σ 分别为样本 x 的均值与方差。

如果 skewness=0，则表示分布形态与正态分布偏度相同；如果 skewness>0，则表示正偏差数值较大，长尾巴拖在右边；如果 skewness<0，则表示负偏差数值较大，为负偏或左偏，长尾巴拖在左边。

峰度的计算公式为

$$\text{kurtisis} = \frac{E(x-\mu)^4}{\sigma^4}$$

正态分布的峰度等于 3、大于 3 表示尖峰，小于 3 表示分布比较均匀。股票市场收益率的时间序列大都为尖峰肥尾。

1. 计算偏度

调用方式

```
Y = skewness(A)
Y = skewness(A,flag)
```

输入参数

A	样本值
flag	偏度的计算方式,0(默认值)为无偏估计,1 为有偏估计

下面是一个例子。

```
>> x = rand(1,20);
>> skewness(x)
ans =
      - 0.0487
```

2. 峰度计算公式

调用方式

```
k = kurtosis(X)
k = kurtosis(X,flag)
k = kurtosis(X,flag,dim)
```

输入参数

X	样本观察矩阵
flag	峰度的计算方式,0(默认)为无偏估计,1 为有偏估计
dim	dim = 1(默认)表示对每列求平均,dim = 2 表示对每行求平均

下面是一个计算峰度的例子。

```
>> x = rand(1,20);
>> kurtosis(x)
ans =
      1.839 2
```

5.2.8 计算绝对离差

绝对离差是以偏差绝对数来衡量决策方案的风险。在期望值相同的情况下,标准离差越大,风险越大;标准离差越小,风险越小。

调用方式

```
Y = mad(X)
Y = mad(X,n)
```

输入参数

X	观察值
n	绝对偏差计算方式 $n=0$（默认）计算公式为 mean(abs(X - mean(X)) $n=1$ 计算公式为 median(abs(X - median(X))

下面是一个计算绝对离差的例子。

```
>> x = rand(1,20);
>> mad(x)
ans =
      0.1593
```

5.2.9 计算中心矩

数理统计中经常用到中心矩的概念，k 阶中心矩的计算公式为

$$m_k = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^k}{n}$$

可以看出 1 阶中心矩为 0，如果观察值是矩阵则以每列为样本计算中心矩。

调用方式

```
M = moment(X,order)
```

输入参数

X	观察样本值
order	中心矩的阶数，必须为正整数

输出参数

```
>> X = randn([6 5])
X =
    -0.4326     1.1892    -0.5883    -0.0956    -0.6918
    -1.6656    -0.0376     2.1832    -0.8323     0.8580
     0.1253     0.3273    -0.1364     0.2944     1.2540
     0.2877     0.1746     0.1139    -1.3362    -1.5937
```

```
   -1.1465   -0.1867    1.0668    0.7143   -1.4410
    1.1909    0.7258    0.0593    1.6236    0.5711
```

下面计算样本的3阶矩。

```
>> m = moment(X,3)
m =
    0.0021    0.0608    0.6710    0.1094   -0.1020
```

5.2.10 计算协方差与相关系数

协方差是一个用于衡量投资组合任意两个资产相关性的统计指标。当协方差为正值时，表示两种资产的收益率呈同方向变动；协方差为负值时，表示两种资产的收益率呈相反方向变化；协方差等于0时，表示两种资产之间不存在相关性。

相关系数总是在－1～1之间的范围内变动，－1表示完全负相关(反向)，1表示完全正相关(同向)，0则表示不相关。MATLAB计算协方差、相关系数的函数分别是cov和corrcoef。

1. 协方差

调用方式

```
C = cov(X)
C = cov(x,y)
```

下面是一个例子。

```
>> A = [-1 1 2 ; -2 3 1 ; 4 0 3];
>> cov(A)
ans =
   10.3333   -4.1667    3.0000
   -4.1667    2.3333   -1.5000
    3.0000   -1.5000    1.0000
```

2. 相关系数

调用方式

```
R = corrcoef(X)
R = corrcoef(x,y)
[R,P] = corrcoef(X,'param1',val1,'param2',val2,...)
```

输入参数

X	观察值矩阵
y	观察向量
param1	参数 1，参数的值如下： alpha 表示置信度，在 0～1 之间
val1	参数 1 的值
param2	参数 2
val2	参数 2 的值

输出参数

R	相关系数矩阵
P	每个相关系数的概率矩阵

下面是一个计算相关系数的例子。

```
>> x = randn(30,4);              % 生成不相关的随机数矩阵
>> x(:,4) = sum(x,2);            % 沿列方向求和
>> [r,p] = corrcoef(x)           % 计算相关系数以及概率(p 值)
>> [i,j] = find(p<0.05);         % 找出 p 值小于 5 % 元素的位置
>> [i,j]
r =
      1.0000    0.2220   -0.1534    0.3460
      0.2220    1.0000    0.1432    0.5000
     -0.1534    0.1432    1.0000    0.6601
      0.3460    0.5000    0.6601    1.0000
p =
    1.0000    0.2385    0.4182    0.0611
    0.2385    1.0000    0.4502    0.0049
    0.4182    0.4502    1.0000    0.0001
    0.0611    0.0049    0.0001    1.0000
ans =
     4     2
     4     3
     2     4
     3     4
```

5.3.2 最小二乘拟合数据

在 MATLAB 中使用最小二乘拟合的命令是 lsline，下面是一个例子。

```
>> x = rand(1,20)
>> x = cumsum(x)
>> plot(x,'+')
>> lsline
```

最小二乘拟合图如图 5.3 所示。

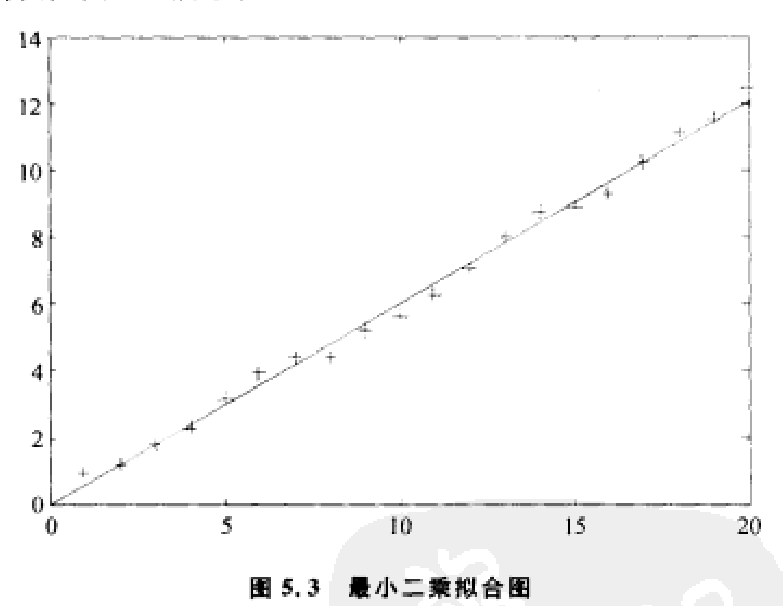


图 5.3 最小二乘拟合图

5.3.3 正态分布概率图

有时需要判断样本数据是否服从正态分布，normplot 函数用图的形式给出直观的判断。如果数据点越靠近直线则分布越近似正态分布，越远则分布越偏离正态分布。

```
>> x = normrnd(0,1,20,1);
>> plot(x,'+')
>> normplot(x)
```

正态分布拟合图如图 5.4 所示。

从图 5.4 可以看出，数据点基本上是直线，所以符合正态分布。如果判断数据是否服从 Weibull 分布则可以对生成的数据用 weilplot 函数进行判断。图 5.5 给出了 Weibull 分布拟合的结果。

从图 5.5 中可以看出对于数据较小、较大的点偏离较大，数据不服从 Weibull 分布。

5.3 统计绘图

5.3.1 样本频率分布图

样本频率分布图函数是 tabulate。

```
>> x=[1 2 3 5 7 3 3.4]
x =
   1.0000    2.0000    3.0000    5.0000    7.0000    3.0000    3.4000
>> tabulate(x)
Value    Count    Percent
     1        1     14.29%
     2        1     14.29%
     3        2     28.57%
   3.4        1     14.29%
     5        1     14.29%
     7        1     14.29%
```

下面调用 cdfplot 函数绘出 x 的分布图。

```
>> cdfplot(x)
```

向量 x 的分布如图 5.2 所示。

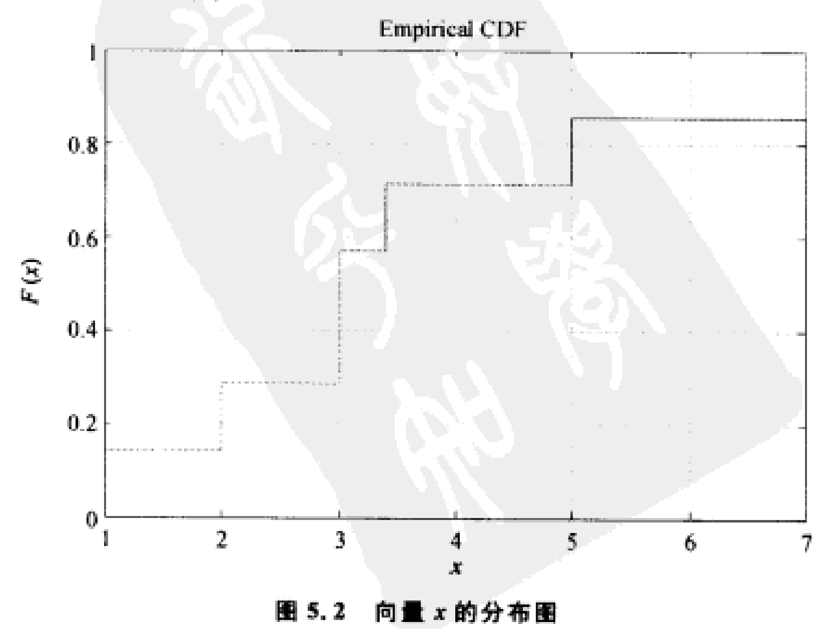


图 5.2 向量 x 的分布图

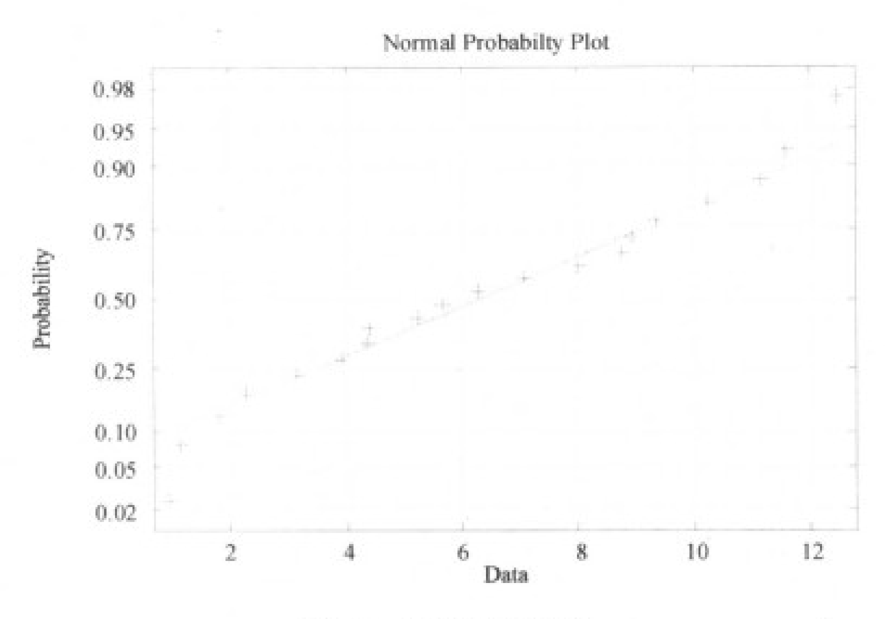


图 5.4　正态分布拟合图

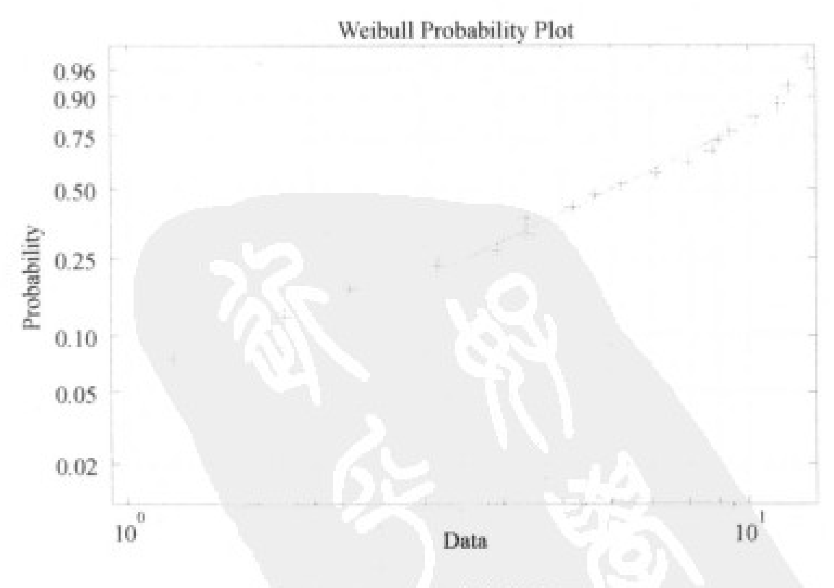


图 5.5　Weibull 分布拟合图

5.3.4　样本密度图

在 MATLAB 中绘出样本数据的密度图函数为 capaplot。

```
>> randn('seed',0);
>> x = randn(1,20);
>> x = cumsum(x)
```

```
>> x = cumsum(x)
>> capaplot(x,[0,10])
```

样本的密度示意图如图 5.6 所示。

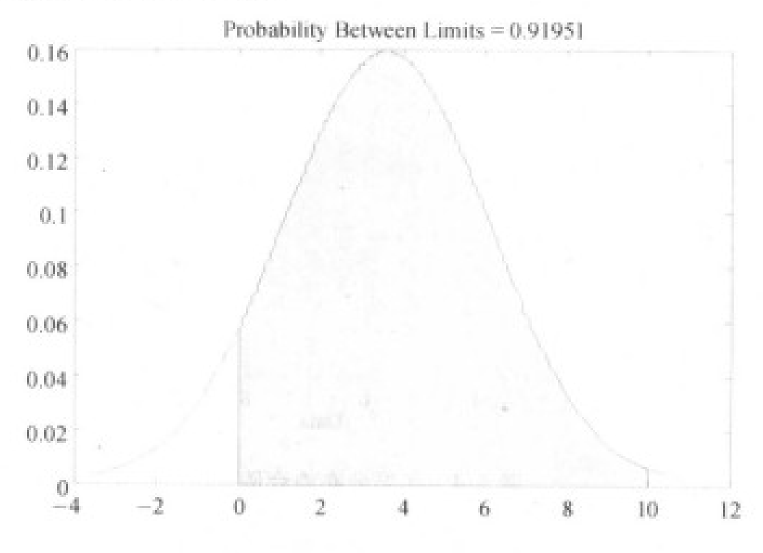


图 5.6 样本的密度示意图

5.3.5 频率直方图

调用方式

```
n = hist(Y)
n = hist(Y,n bins)
[n,xout] = hist(Y,n,bins)
```

输入参数

Y	观察值。如果 **Y** 是一个向量，则画出一个频率图；如果 **Y** 是一个 $m \times p$ 阶矩阵，则对 **Y** 每一列分别作频率图
nbins	频率图分成 nbins 等分的区间段，默认值为 10

输出参数

n	样本落在区间段的频率
xout	区间段的刻度

下面是一个例子。

```
>> x = randn(1,20)
>> randn('seed',0)
```

```
>> x = randn(1,200);
>> hist(x)
```

频率直方图如图 5.7 所示。

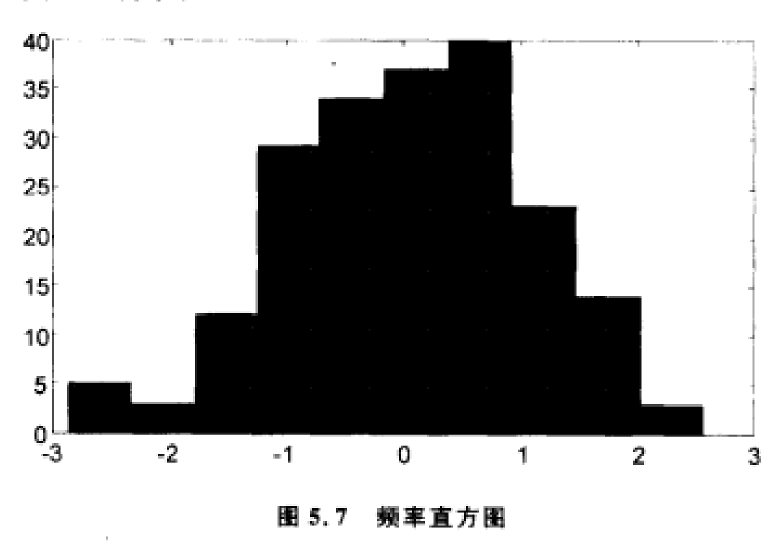


图 5.7 频率直方图

```
>> hist(x,min(x):0.3:max(x));
```

其中 min(x):0.3:max(x)表示频率图 X 轴的刻度起点是 x 最小元素，终点是 x 中最大元素，刻度间隔 0.3。规定刻度间隔的频率直方图如图 5.8 所示。

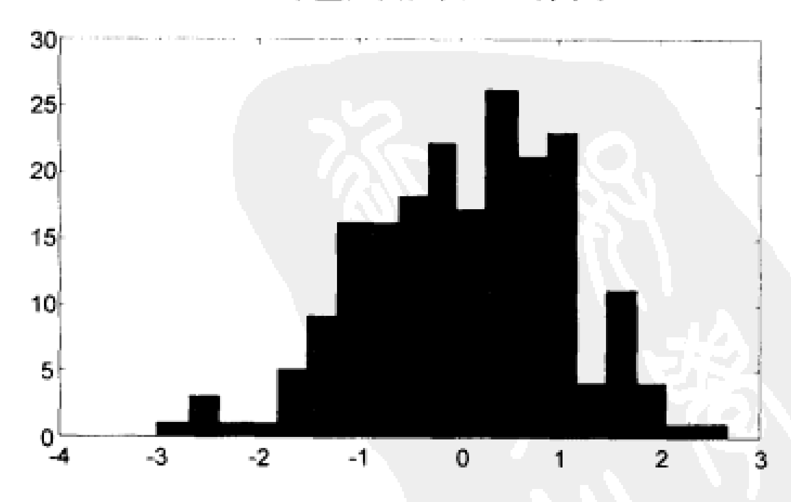


图 5.8 规定刻度间隔的频率直方图

如果在频率图的基础上加上正态分布拟合图，则可以用 histfit 函数。

```
>> randn('seed',0)
>> x = randn(1,20)
>> histfit(x)
```


带有密度函数的频率直方图如图 5.9 所示。

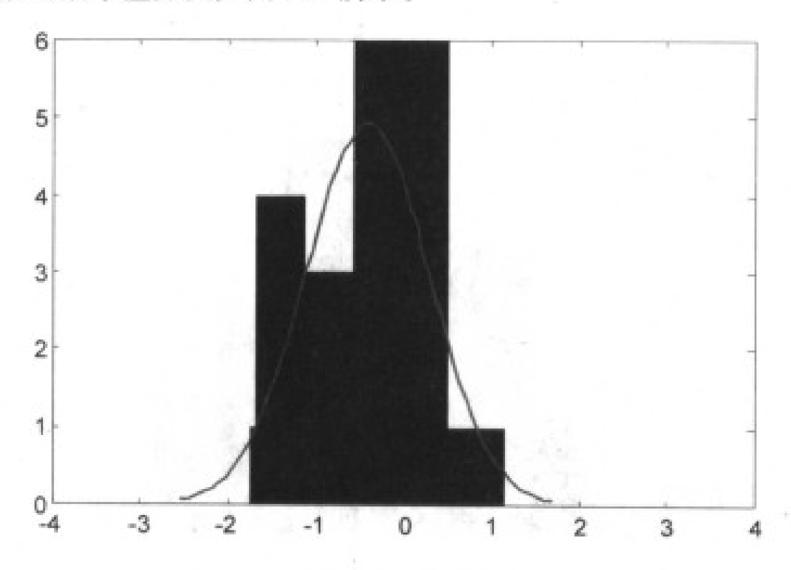


图 5.9　带有密度函数的频率直方图

5.3.6　盒　图

在 MATLAB 中绘制样本数据的盒图函数是 boxplot，用法如下：

调用方式

```
boxplot(X)
boxplot(X,G)
boxplot(X,'Param1', val1, 'Param2', val2,...)
```

输入参数

X	样本观察值
G	各组的名称
Param1	参数 1 的名称
val1	参数 1 的值
	各参数的名称和内容如表 5.2 所列

表 5.2　boxplot 函数中各参数内容

参数名称	参数值
notch	'on'生成有缺口的盒图，'off'生成矩形盒图
symbol	线条类型，默认值为'r+'
orientation	'vertical'（默认值）为垂直型盒图，'horizontal'为水平型盒图

续表 5.2

参数名称	参数值
whisker	盒图须线的长度，默认值=1.5×四分位间距
labels	盒图行列的名称标签
colors	线条颜色
widths	盒图的宽度，默认值=0.5
positions	盒图的位置，默认为 1 : n
grouporder	组的次序

```
>> x1 = normrnd(5,1,100,1);
>> x2 = normrnd(6,1,100,1);
>> boxplot([x1,x2],'notch','on')
```

正态分布的盒图如图 5.10 所示。

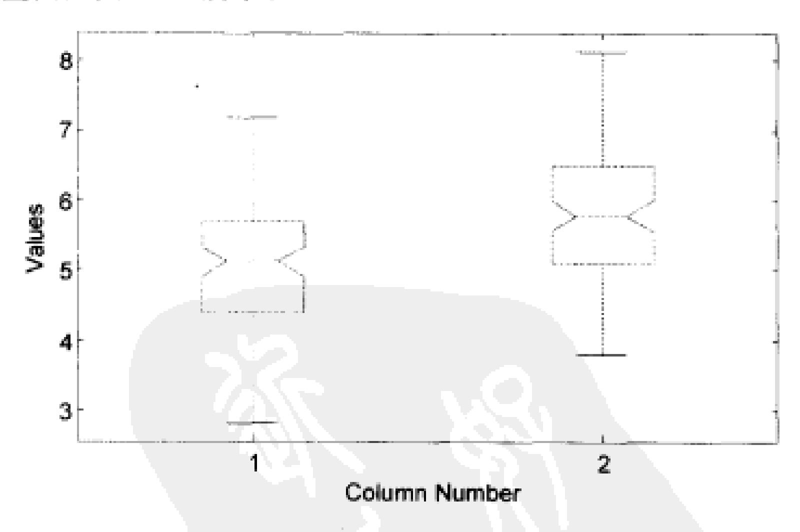


图 5.10 正态分布的盒图

图 5.10 的内容说明如下：

- 盒子的上下两条线分别为样本的 25 %和 75 %分位数，盒子的上下底之间的距离为四分位的间距。
- 盒子的中间线为样本中值，如果样本中值不在盒子的中间，表示存在一定的偏度。盒子的上方、下方有两条虚线，显示了样本值的范围，野值(异常值)位于超过盒子顶端、底端 1.5 倍的四分位数。
- 含有缺口的盒图中齿形缺口表示样本中值的置信区间。

5.4 多元线性回归分析

在金融上常常需要对金融、经济数据进行回归，其中最简单的是多元线性回归。

5.4.1 多元线性回归

假设因变量 $\boldsymbol{Y}$ 和自变量 $\boldsymbol{X}$ 之间服从以下的线性模型。

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \epsilon$$

式中：$\boldsymbol{Y}$ 是因变量的观察值，$\boldsymbol{X}$ 是自变量回归矩阵，β 是参数向量，ϵ 是白噪声。

β 的最小二乘解是

$$\beta = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y$$

调用方式

```
b = regress(Y,X)
[b,bint] = regress(Y,X)
[b,bint,r] = regress(Y,X)
[b,bint,r,rint] = regress(Y,X)
[b,bint,r,rint,stats] = regress(Y,X)
[b,bint,r,rint,stats] = regress(Y,X,alpha)
```

输入参数

X	自变量观察值，注意如果模型中有常数项，则 $\boldsymbol{X}$ 的第一列所有元素为 1
Y	因变量观察值向量
alpha	参数的置信度

输出参数

b	β 的估计值，注意 b 中已经包含了常数项
bint	β 的置信区间
r	残差
rint	残差的置信区间
stats	R^2、F、概率 p

【例 5-1】 首先按照下面模型生成一系列随机数，然后回归。

$$Y = 0.1 + 0.4X + \epsilon$$
$$\epsilon \sim N(0,0.1)$$

下面生成一组随机数。

```
>> X = 1:10;
```

```
>> Y = 0.1 + 0.4 * X + normrnd(0,0.1,1,10);
```

下面估计 β。

```
>> [b,bint,r,rint,stats] = regress(Y',[ones(10,1),X'],0.05) %
b =
    0.0213
    0.4143
bint =
       - 0.1111      0.1537
         0.3930      0.4357
r =
         0.0211
       - 0.1165
         0.0482
         0.0501
       - 0.1076
         0.1118
         0.0973
       - 0.0397
       - 0.0175
       - 0.0471
rint =
       - 0.1455      0.1877
       - 0.2647      0.0316
       - 0.1351      0.2316
       - 0.1384      0.2387
       - 0.2800      0.0648
       - 0.0585      0.2821
       - 0.0767      0.2713
       - 0.2246      0.1452
       - 0.1965      0.1614
       - 0.2097      0.1154
stats =
         1.0e + 003 *
         0.0010      2.0047      0.0000      0.0000
```

从 b 的估计值可以得知常数项和一次项的系数分别为 0.021 3,0.414 3。在 0.05 置信水平下常数项系数估计区间为[−0.111 1　0.153 7],$\boldsymbol{X}$ 的系数置信区间为[0.393 0　0.435 7]。

由于样本数量非常少，参数估计并不稳定。图 5.11 是残差及其置信区间图。

```
>> rcoplot(r,rint)
```

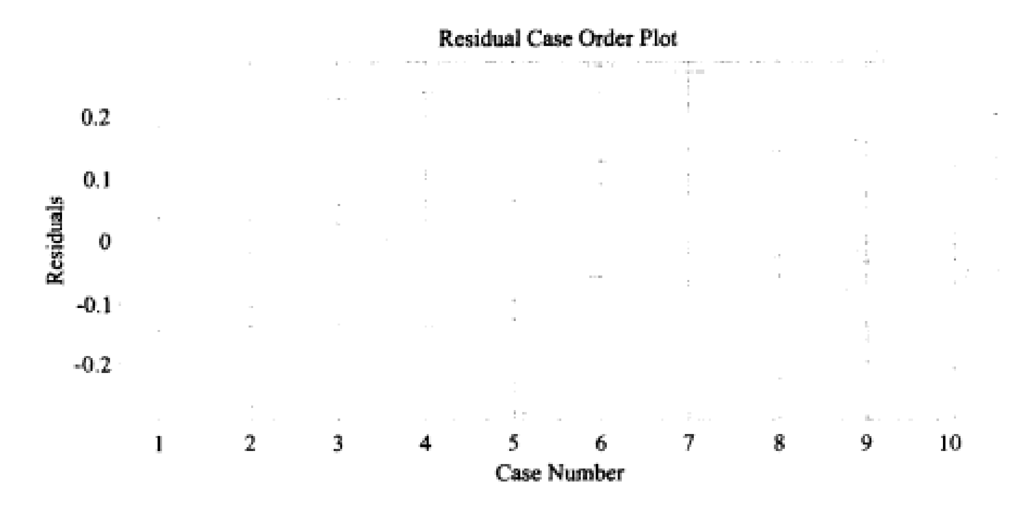


图 5.11　残差和置信区间图

5.4.2　多元正态回归

在 MATLAB 中 mvnrmle 函数可以进行多元正态回归，假设 Y_k 为随机变量，其分布如下：

$$Y_k \sim N(\mathrm{Desingn}_k \times \mathrm{Parameters}, \mathrm{Covariance})$$

式中：$N(g,g)$为多元正态分布。

调用方式

```
[Parameters, Covariance, Resid, Info] = mvnrmle(Y, Design, MaxIterations, TolParam, TolObj, Covar0)
```

输入参数

Y	观察值矩阵，$Y_{n\times k}$中 n 是样本的个数，k 是资产的数目
Design	自变量单元变量矩阵，如果 **Y** 只有一个资产时，Design 是一个矩阵，如果 **Y** 中的资产个数大于一个时，Design 是一个单元向量，每个元素都是一个矩阵。**Y** 的 k 列和 Design 第 k 个元素中的矩阵进行回归
MaxIterations	
TolParam	
TolObj	
Covaro	

输出参数

Parameters	参数
Covariance	协方差
Resid	残差
Info	估计过程的相关信息

5.4.3　估计多元正态分布每个资产的标准差

调用方式

```
[StdParameters, StdCovariance] = mvnrstd(Data, Design, Covariance)
```

输入参数

Data	观察值矩阵，$Y_{n\times k}$中 n 是样本的个数，k 是资产的数目
Design	自变量单元变量矩阵，如果 Y 只有一个资产时，Design 是一个单元向量，如果 Y 含有多于一个资产时，Design 是一个单元向量。
Covariance	回归时的残差

输出参数

StdParameters	每个资产的标准差
StdCovariance	协方差

5.4.4　岭回归

线性回归中参数估计 $\beta=(X^{T}X)^{-1}X^{T}Y$，如果观察值 X 存在自相关性，则$(X^{T}X)$是奇异矩阵，估计值就会出现非常大的误差，这时矩阵 $X^{T}X$ 需要加上一个对角元素是常数 k 的单位阵，即 $\beta=(X^{T}X+kI)^{-1}X^{T}Y$。MATLAB 提供了岭回归 ridge 函数求解该问题。

调用方式

```
b1 = ridge(Y,X,k)
b0 = ridge(Y,X,k)
```

输入参数

Y	因变量观察值
X	自变量观察值
k	k 表示控制系数，可以根据需要进行选择。

输出参数

b	模型估计参数

【例 5－2】 对 acetylene 文件中的数据进行岭回归。

```
>> load acetylene
```

查看工作区中的变量。

```
>> who
   Your variables are:
   Description  x1        x2        x3        y
```

自变量是 $x1$、$x2$、$x3$，因变量是 y。

下面画出变量之间折线图。

```
>> subplot(1,3,1)
>> plot(x1,x2,'.')                % x1,x2 折线图
>> xlabel('x1'); ylabel('x2'); grid on; axis square
>> subplot(1,3,2)
>> plot(x1,x3,'.')                % x1,x3 折线图
>> xlabel('x1'); ylabel('x3'); grid on; axis square
>> subplot(1,3,3)
>> plot(x2,x3,'.')                % x2,x3 折线图
>> xlabel('x2'); ylabel('x3'); grid on; axis square
```

图 5.12 画出了不同变量之间的折线图。

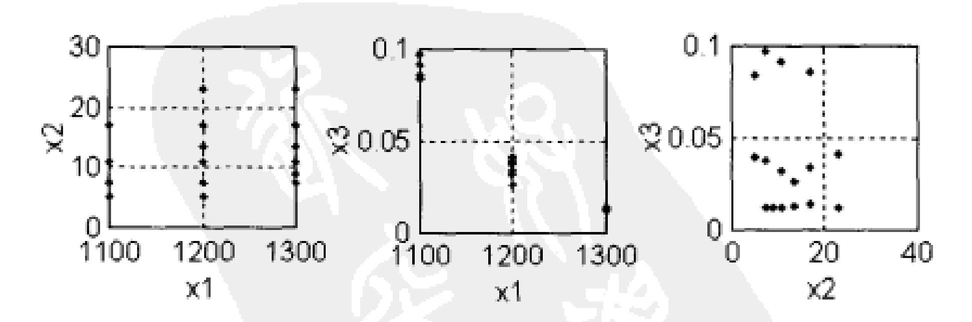


图 5.12　不同变量之间的折线图

从图 5.12 的图中可以看出 $x1$、$x2$ 可能存在共线性，$x1$、$x3$ 可能存在共线性。

```
>> X = [x1 x2 x3];
>> D = [x1,x2,x3,x1.*x2,x1.*x3,x2.*x3];        % 生成新矩阵
```

下面进行不含常数项的岭回归：

```
>> k = 0:1e-5:5e-3;
>> b = ridge(y,D,k);
```

画出 k 与 b 之间收敛情况。

```
>> figure
>> plot(k,b,'LineWidth',2)                    % 画出 k 与 b 之间的相关性
>> ylim([-100 100])
>> grid on
>> xlabel('参数 k')
>> ylabel('回归系数')
>> title('{\bf 岭回归参数与 k 之间关系}')
>> legend('x1','x2','x3','x1x2','x1x3','x2x3')
```

图 5.13 画出了岭回归的收敛情况。

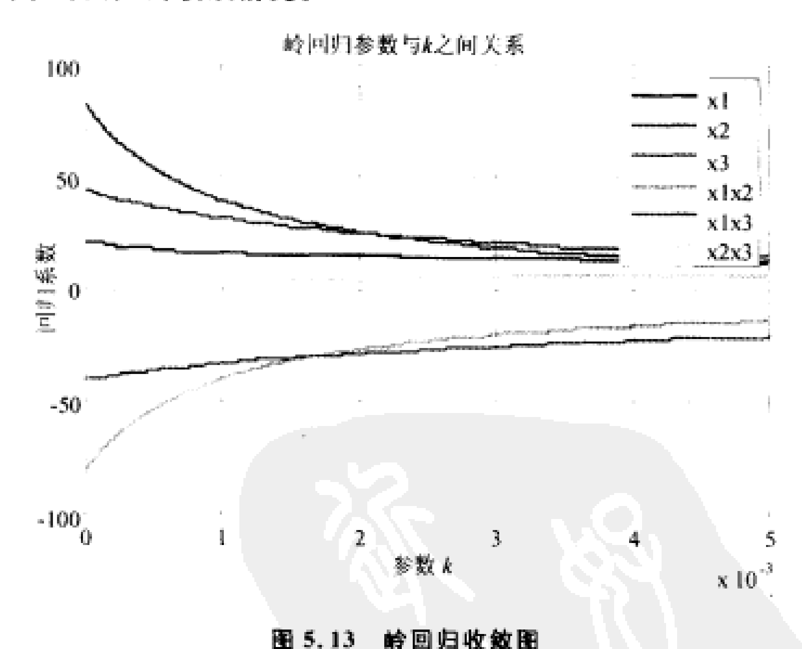


图 5.13　岭回归收敛图

5.5　主成分分析

主成分分析是在各个变量之间相关关系研究的基础上，用较少的新变量代替原来较多的变量，而且使这些较少的新变量尽可能多地保留原来较多的变量所反映的信息。

5.5.1　主成分分析基本原理

设样本为 $X_1, X_2, X_3, \cdots, X_p$，其对应的样本均值为 $\overline{X_1}, \overline{X_2}, \overline{X_3}, \cdots, \overline{X_p}$，对应的标准差为 $S_1, S_2, S_2, \cdots, S_p$。

首先对样本进行标准化处理，为简单起见，标准化后的样本仍记为 $X_1, X_2, X_3, \cdots, X_p$。

主成分具有如下性质：

设 $F_1, F_2, F_3, \cdots, F_p$ 是主成分，也即是 $X_1, X_2, X_3, \cdots, X_p$ 的线形表示，同时满足如下条件：

① 主成分是原样本的正交变换。

② 各主成分之间互不相关。

③ 主成分的总方差不变。

④ 主成分按方差从大到小排序。

这一性质说明，主成分是原变量的线性组合，是对原变量信息的一种改良；主成分不增加总信息量，也不减少总信息量。

保留多少个主成分取决于保留部分的累积方差在方差总和中所占百分比（即累计贡献率），它标志着前几个主成分概括信息的多寡。在实践中，粗略规定一个百分比便可决定保留几个主成分，如果多留一个主成分，但累积方差增加无几，便不再多留。

设 $\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_p$ 为主成分的特征值，前 k 个方差累积贡献率为

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\sum \lambda_i}$$

一般当累积贡献率大于 85 %时不再增加新的主成分。

5.5.2 主成分分析函数

在 MATLAB 中提供了两个主成分分析函数 princomp 和 pcacov。

调用方式

```
[COEFF, SCORE] = princomp(X)
[COEFF, SCORE,latent] = princomp(X)
[COEFF, SCORE,latent,tsquare] = princomp(X)
```

输入参数

X	观察变量

输出参数

COEFF	主成分系数
SCORE	新坐标系
latent	**X** 的协方差矩阵的特征值
tsquare	Hotelling 统计量的值

【例 5 - 3】 用 MATLAB 自带数据进行主成分分析。MATLAB 中自带了数据文件

hald，可以直接调用进行主成分分析。hald数据考虑了影响温度的4个因素，温度保存在heat变量中，4个因素的观察值保存在ingredients中。由于4个因素之间存在相关性，无法直接回归，为解决这个问题，首先进行主成分分析，生成4个主成分变量，主成分之间互不相关，而且和观察值的信息是同样的。

第一步：载入数据，考查变量之间的相关性。

```
>> load hald                    % 载入 MATLAB 自带的数据文件
```

考查相关性。

```
>> corrcoef(ingredients)
ans =
        1.0000      0.2286     -0.8241     -0.2454
        0.2286      1.0000     -0.1392     -0.9730
       -0.8241     -0.1392      1.0000      0.0295
       -0.2454     -0.9730      0.0295      1.0000
```

发现自变量2与变量3之间高度相关，所以需要剔除相关性。

第二步：主成分分析。

```
>> [pc,score,latent,tsquare] = princomp(ingredients)
```

主成分系数

```
pc =
        0.0678      0.6460     -0.5673      0.5062
        0.6785      0.0200      0.5440      0.4933
       -0.0290     -0.7553     -0.4036      0.5156
       -0.7309      0.1085      0.4684      0.4844
```

主成分的方差贡献率

```
score =
          -36.8218      6.8709      4.5909      0.3967
          -29.6073     -4.6109      2.2476     -0.3958
           12.9818      4.2049     -0.9022     -1.1261
          -23.7147      6.6341     -1.8547     -0.3786
            0.5532      4.4617      6.0874      0.1424
           10.8125      3.6466     -0.9130     -0.1350
           32.5882     -8.9798      1.6063      0.0818
          -22.6064    -10.7259     -3.2365      0.3243
            9.2626     -8.9854      0.0169     -0.5437
```

```
    3.2840    14.1573    -7.0465    0.3405
   -9.2200   -12.3861    -3.4283    0.4352
   25.5849     2.7817     0.3867    0.4468
   26.9032     2.9310     2.4455    0.4116
```

协方差的特征值

```
latent =
  517.7969
   67.4964
   12.4054
    0.2372
tsquare =
    5.6803
    3.0758
    6.0002
    2.6198
    3.3681
    0.5668
    3.4818
    3.9794
    2.6086
    7.4818
    4.1830
    2.2327
    2.7216
```

由此可以得到4个主成分如下：

$$y_1=0.0678x_1+0.06785x_2-0.029x_3-0.0731x_4$$

$$y_2=0.6460x_1+0.0200x_2-0.7553x_3+0.1085x_4$$

$$y_3=-0.5673x_1+0.5440x_2-0.4036x_3+0.4684x_4$$

$$y_4=0.5062x_1+0.4933x_2+0.5156x_3+0.4844x_4$$

从特征值可以看出前面两个主成分可以很好地解释98 %的方差。

(517.796 9+67.496 4)/(517.7969+67.496 4+12.405 4+0.237 2)=98 %

采用pcacov函数计算主成分的结果同princomp函数的结果是一样的。

```
>> load hald
```

然后计算观察变量ingredients协方差。

```
>> covx = cov(ingredients);
>> [pc,latent,explain] = pcacov(covx)
```

在确定主成分后对主成分进行回归。
将生成4个主成分,保存在变量 new 中,变量 new 中的每列就是一个主成分。

```
>> new = ingredients * pc;
```

验证主成分之间的相关性:

```
>> corrcoef(new)
ans =
      1.0000      0.0000      0.0000      0.0000
      0.0000      1.0000      0.0000      0.0000
      0.0000      0.0000      1.0000      0.0000
      0.0000      0.0000      0.0000      1.0000
```

发现主成分之间是不相关的。
第三步:用前两个主成分进行回归。

```
>> regress(heat,new(:,1:2))
ans =
      2.1843
      1.0894
```

这样温度和主成分之间的关系如下:

$$\text{heat} = 2.1843y_1 + 1.0894y_2 + \epsilon_t$$

还原成线性回归形式:

```
>> 2.1843 * pc(:,1) + 1.0894 * pc(:,2)
ans =
        0.8519
        1.5039
      - 0.8862
      - 1.4783
```

这样温度和自变量之间的关系如下:

$$\text{heat} = 0.8519x_1 + 1.5039x_2 - 0.8862x_3 - 1.4783x_4 + \epsilon_t \tag{5.1}$$

第四步:验证主成分分析优点。
下面计算 heat 和 ingredients 中各个分量的相关系数。

```
>> corr(heat,ingredients(:,1))
```

```
ans =
      0.7307
>> corr(heat,ingredients(:,2))
ans =
     0.8163
>> corr(heat,ingredients(:,3))
ans =
      -0.5347
>> corr(heat,ingredients(:,4))
ans =
      -0.8213
```

相关系数表明前两个自变量和因变量之间是正相关,后两个自变量和因变量之间负相关。式(5.1)和实际结果相印证。

如果不用主成分回归,而是直接对自变量和因变量进行回归。

```
>> c0 = regress(heat,ingredients)
c0 =
    2.1930
    1.1533
    0.7585
    0.4863
```

自变量和因变量之间的函数关系式如下:

$$heat = 2.1930x_1 + 1.1533x_2 + 0.7585x_3 + 0.4863x_4 + \varepsilon \tag{5.2}$$

从式(5.2)可以看出自变量和因变量之间都是正相关,显然和实际结果不相印证。

5.6 因子分析

因子分析法(factor analysis)是一种用来分析隐藏在表象背后的因子作用的一类统计模型和方法,它起源于心理度量学,最初是研究如何用少数几个变量来解释众多原始变量,同时又尽量避免信息丢失的多元统计分析方法。在实际问题的分析过程中,常用因子分析去除重叠信息,将原始的众多指标综合成较少的几个因子变量来分析。

1. 因子分析法相关概念

设样本为 $X_1, X_2, X_3, \cdots, X_p$,其对应的样本均值为$\overline{X_1}, \overline{X_2}, \overline{X_3}, \cdots, \overline{X_p}$,对应的标准差为 $S_1, S_2, S_3, \cdots, S_p$。

首先对样本进行标准化处理，为简单起见，标准化后的样本仍记为 $X_1, X_2, X_3, \cdots, X_p$。

因子分析的数学模型如下：

$$X = fX + s$$

式中：X 是经过正交处理的样本值，f 是负荷矩阵，表示公共因子部分，s 是其他特殊因子。要求 f，s 之间不相关而且不可观察。

因子分析的核心问题有两个，一是确定因子的个数，二是对因子变量进行命名。

2. 因子分析函数调用方式

调用方式

```
lambda = factoran(X,m)
[lambda,psi] = factoran(X,m)
[lambda,psi,T] = factoran(X,m)
[lambda,psi,T,stats] = factoran(X,m)
[lambda,psi,T,stats,F] = factoran(X,m)
```

输入参数

X	观察值矩阵，每列属于同一个变量
m	公共因素的个数

输出参数

lambda	负荷矩阵估计值
psi	特殊因素矩阵估计值
T	因素负荷旋转矩阵
stats	假设检验
F	**F** 是 $n \times m$ 维因子的分矩阵

下面用一个例子说明。

【例 5-4】 在 MATLAB 中自带了 10 个公司 100 周的收益率数据，10 家公司分成 3 个行业，4 家属于科技公司，3 家属于金融公司，另外 3 家属于零售类公司。一般来讲，同一行业上市公司同质性强，走势也相同，不同行业之间股票受不同风险因素影响，走势差别较大。下面验证该结论。

```
>> load stockreturns;
>> m = 3;                  % 考虑 3 个行业的因素
>> [LoadingsPM,specificVarPM] = factoran(stocks,m,'rotate','promax')
LoadingsPM =
            0.9452      0.1214      -0.0617
```

```
          0.7064     -0.0178      0.2058
          0.3885     -0.0994      0.0975
          0.4162     -0.0148     -0.1298
          0.1021      0.9019      0.0768
          0.0873      0.7709     -0.0821
         -0.1616      0.5320     -0.0888
          0.2169      0.2844      0.6635
          0.0016     -0.1881      0.7849
         -0.2289      0.0636      0.6475
```

从上述结果明显可以看出第1到第4家公司属于同一类公司，第5到第7家属于同一类公司，第8到第10家属于同一类公司，说明不同行业间的股票收益率存在不同。

```
specificVarPM =
              0.0991
              0.3431
              0.8097
              0.8559
              0.1429
              0.3691
              0.6928
              0.3162
              0.3311
              0.6544
```

5.7 方差分析

方差分析主要用来检验两个以上样本的平均值差异的显著程度，由此判断样本究竟是否抽自具有同一均值的总体。

5.7.1 单因素方差分析

在方差分析中，通常把对试验结果发生影响和起作用的自变量称为因素。如果分析一个因素对于试验结果的影响和作用，就称为单因素方差分析。

一般地，假定所检验的结果受某一因素 A 的影响，可以取 k 个不同的水平 $1,2,3,\cdots,k$。对于因素的每一个水平 i 都进行 n 次试验，结果分别为 $X_{i1},X_{i2},\cdots,X_{in}$，一般把这一组样本记作 X_i。假定 $X_i \sim N(\mu_i,\sigma^2)$，即对于因素的每一个水平，所得到的结果都服从正态分布，且方差相等。

5.7.2 方差分析步骤

方差分析一方面确定因素的不同水平下均值之间的方差，作为所有试验数据所组成的全部总体方差的一个估计值；另一方面，再考虑同一水平下不同试验数据对于这一水平的均值的方差。由此，计算出全部数据的总体方差的第二个估计值；最后，比较上述两个估计值。如果这两个方差的估计值比较接近就说明因素的差异并不大，则接受零假设；否则接受备择假设。根据上述思路可以得到方差分析的方法和步骤。

1. 提出假设

H_0：$\mu_1=\mu_2=\cdots=\mu_k$，即因素的不同水平对试验结果无显著影响。

H_1：不是所有的 μ_i 都相等($i=1,2,\cdots,k$)，即因素的不同水平对试验结果有显著影响。

2. 方差分解

下面先定义总离差平方和为各样本观察值与总均值的离差平方和。

记

$$\mathrm{SST}=\sum_{i=1}^{k}\sum_{j=1}^{n}(X_{ij}-\overline{X})^2$$

式中：$\overline{X}$是样本总均值，即 $\overline{X}=\dfrac{(\sum\limits_{i=1}^{k}\sum\limits_{j=1}^{n}X_{ij})}{N}$；$N=nk$ 为样本观察值总数。

将总离差平方和分解为两部分：

$$\mathrm{SST}=\sum_{i=1}^{k}\sum_{j=1}^{n}(X_{ij}-\overline{X})^2=\sum_{i=1}^{k}\sum_{j=1}^{n}[(X_{ij}-\overline{X}_i)+(\overline{X}_i-\overline{X})]^2=$$
$$\sum_{i=1}^{k}\sum_{j=1}^{n}(X_{ij}-\overline{X}_i)^2+\sum_{i=1}^{k}n\cdot(X_i-\overline{X})^2$$

式中：$\overline{X}_i$ 是第 i 个样本的平均值，即

$$\overline{X}_i=\frac{(\sum\limits_{j=1}^{k}X_{ij})}{n}$$

记

$$\mathrm{SSE}=\sum_{i=1}^{k}\sum_{j=1}^{n}(X_{ij}-\overline{X}_i)^2$$

表示同一样本组内，由于随机因素影响所产生的离差平方和，简称为组内平方和。

记

$$\text{SSR} = \sum_{i=1}^{k} n \cdot (X_i - \overline{X})^2$$

表示不同的样本组之间，由于变异因素的不同水平影响所产生的离差平方和，简称为组间平方和。

由此可以得到

$$\text{SST} = \text{SSR} + \text{SSE}$$

对应于 SST、SSR 和 SSE 的自由度分别为 $n-1$、$k-1$、$n-k$。

3. F 检验

将 SSR 和 SSE 分别除以其自由度，即得各自的均方差：

组间均方差　MSR=SSR/$(k-1)$

组内的均方差　MSE=SSE/$(n-k)$

检验统计量

$$F = \frac{\text{MSR}}{\text{MSE}}$$

当原假设 $H_0: \mu_1 = \mu_2 = \cdots = \mu_k$ 成立时，$E(\text{MSE}) = E(\text{MSR}) = \sigma^2$。此时 MSR 较小，$F$ 值也较小。反之 H_0 不成立时，MSR 较大，F 值也较大。对于给定的显著性水平 α 查 F 分布表得到 $F_{1-\alpha}(k-1,n-k)$。如果 $F > F_{1-\alpha}(k-1,n-k)$，则原假设不成立，即 k 个组的总体均值之间有显著的差异，就拒绝 H_0。若 $F \leqslant F_{1-\alpha}(k-1,n-k)$，则原假设成立，即 k 个组的总体均值之间没有显著的差异，就接受 H_0。

4. 方差分析表

上述方差分析的方法可以用一张标准形式的表格来实现，称为方差分析表。方差分析表分为五列：第一列表示方差的来源；第二列表示方差的离差的平方和；第三列表示自由度；第四列表示均方差；第五列表示统计检验量 F。表格又分为三行：第一行是组间的方差 SSR 和均方差 MSR，表示因素的不同水平的影响所产生的方差，其值作为计算统计检验量 F 时的分子；第二行是组内方差 SSE 和均方差 MSE，表示随机误差所引起的方差，其值作为计算统计检验量 F 的分母；第三行是检验行，表示总的方差 SST。

由于方差分析表（见表 5.3）概括了方差分析的统计量之间的关系，在进行方差分析时就可以直接按照方差分析表来逐行、逐列地计算出有关的统计量，最后得到检验量 F 的值，并把这一 F 值与查表所得到的一定显著性水平下的 F 检验的临界值进行比较，然后做出接受或拒绝原假设的结论。

表 5.3 单因素方差分析表

方差来源	离差平方和	自由度	均方差	统计检验量 F
组间差	SSR	$k-1$	MSR	$F=\frac{MSR}{MSE}$
组内差	SSE	$n-k$	MSE	
总方差	SST	$n-1$		

5.7.3 单因素方差分析函数

单因素方差分析是比较两组和多组样本的均值，假设各组变量之间相互独立，方差相等，且服从正态分布。原假设是各组均值全部相等。

调用方式

```
p = anova1(X)
p = anova1(Y,group)
p = anova1(Y,group,'displayopt')
[p,table] = anova1(Y,group,'displayopt')
[p,table,stats] = anova1(Y,group,'displayopt')
```

输入参数

X	样本观察值，要求各列为彼此独立的样本
Y	观察值向量
group	组别，**Y** 中每个元素所属的类别
displayopt	取值为"off"与"on"，分别表示掩藏与显示方差分析表图和盒图。盒图上下线为 25 % 和 75 % 分位数，中间线为中位数

输出参数

p	各列均值相等的概率
table	方差分析表
stats	统计量

【例 5-5】 恒瑞医药(600276)和 ST 达声(000007)的股价如表 5.4 所列。

表 5.4 恒瑞医药和 ST 达声价格收益率

ST 达声			恒瑞医药		
日 期	价 格	收益率	日 期	价 格	收益率
2006-8-14	2.487		2006-8-14	13.191	

续表 5.4

ST达声			恒瑞医药		
日　期	价　格	收益率	日　期	价　格	收益率
2006－8－15	2.41	－0.030 96	2006－8－15	13.267	0.005 762
2006－8－16	2.5	0.0373 44	2006－8－17	13.066	－0.015 15
2006－8－17	2.48	－0.008	2006－8－18	13.167	0.007 73
2006－8－18	2.43	－0.020 16	2006－8－21	13.196	0.002 202
2006－8－21	2.41	－0.008 23	2006－8－22	13.196	0
2006－8－22	2.4	－0.004 15	2006－8－23	13.309	0.008 563
2006－8－23	2.4	0	2006－8－24	14.185	0.065 82
2006－8－24	2.37	－0.012 5	2006－8－25	14.143	－0.002 96
2006－8－25	2.42	0.021 097	2006－8－28	14.149	0.000 424
2006－8－28	2.48	0.024 793	2006－8－29	14.285	0.009 612
2006－8－29	2.49	0.004 032	2006－8－30	14.285	0

下面检验二者的收益率是否相等。

```
>> rate = [ - 0.03096      0.037344      - 0.008       - 0.02016      - 0.00823     - 0.00415
               0         - 0.0125        0.021097      0.024793       0.004032      0.005762
           - 0.01515       0.00773       0.002202      0              0.008563      0.06582
           - 0.00296       0.000424      0.009612      0]
>> group = {'stds','stds','stds','stds','stds','stds','stds','stds','stds','stds','stds','hryy','hryy','hryy','hryy','hryy','hryy','hryy','hryy','hryy','hryy','hryy'}
>> [p,table,stat] = anova1(z,group,'on')
p =
     0.4214
table =
    'Source'    'SS'                 'df'     'MS'              'F'          'Prob>F'
    'Groups'    [2.8180e - 004]      [1]      [2.8180e - 004]   [0.6739]     [0.4214]
    'Error'     [        0.0084]     [20]     [4.1815e - 004]   []           []
    'Total'     [        0.0086]     [21]     []                []           []
stat =
    gnames: {2x1 cell}
         n: [11 11]
    source: 'anova1'
     means: [2.9682e - 004 0.0075]
        df: 20
s: [0.0204214]
```

从结果可知 $p=0.4214>0.05$，所以不能拒绝原假设，可以认为二者在此期间的收益率相等。其原因可能是以日收益率为样本，间隔时间太短，不足以反应股票之间的差别。

ST 达声和恒瑞医药方差分析表如图 5.14 所示。

Figure 5: One-way ANOVA

File Edit View Insert Tools Desktop Window Help

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	0.00028	1	0.00028	0.67	0.4214
Error	0.00836	20	0.00042		
Total	0.00864	21			

图 5.14　ST 达声和恒瑞医药方差分析表

从图 5.14 中可以看出方差分析表。

第一列显示数据源(source)。

第二列显示各类数据的平方和(SS)。

第三列显示各类数据相应的自由度(df)。

第四列显示均方值 MS。

第五列显示 Friedman 检验的 F 统计量(F)。

第六列显示 F 统计量对应的概率值。

ST 达声和恒瑞医药收益率盒图如图 5.15 所示。

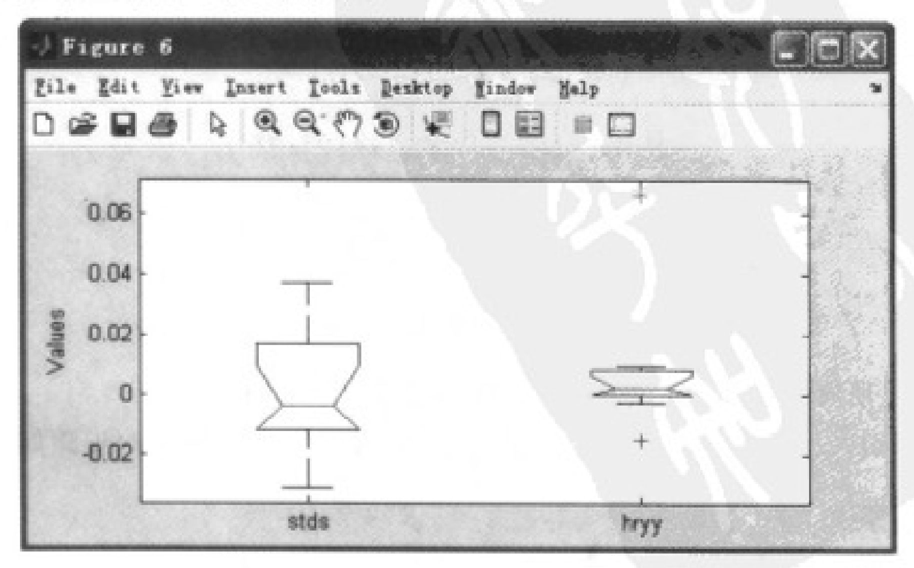


图 5.15　ST 达声和恒瑞医药收益率盒图

5.7.4　双因素方差分析

前面所研究的是试验结果仅受一个因素影响的情形。要求检验的是当因素取不同水平时对结果所产生的影响是否显著。但在实践中，某种试验结果往往受到两个或两个以上因素的影响。双因素方差分析的基本思想与单因素方差分析基本相同。首先分别计算出总变差、各个因素的变差以及随机误差的变差；其次根据各变差相应的自由度求出均方差；最后计算出 F 值并作 F 检验。

设因素 A 有 a 个水平，因素 B 有 b 个水平，试验的重复次数记作 n。记 X_{ijk} 为在因素 A 的第 i 个水平、因素 B 的第 j 个水平下进行第 k 次试验时的观察值($i=1,2,\cdots,a;j=1,2,\cdots,b;k=1,2,\cdots,n$)。记

$$(AB)_{ij} = \sum_{j=1}^{n} X_{ijk}$$

为在因素 A 的第 i 个水平、因素 B 的第 j 个水平下进行各次重复试验的所有观察值的总和。记

$$(\overline{A}\,\overline{B})_{ij} = \frac{(AB)_{ij}}{n} = \frac{1}{n}\sum_{j=1}^{n} X_{ijk} \quad (i = 1,2,\cdots,a;j = 1,2,\cdots,b)$$

为在因素 A 的第 i 个水平、因素 B 的第 j 个水平下进行各次重复试验的所有观察值的平均值。记

$$\mathrm{A}_i = \sum_{j=1}^{b} (AB)_{ij}$$

$$\overline{A_i} = \frac{1}{nb}A_i \quad (i = 1,2,\cdots,a)$$

$$\mathrm{B}_j = \sum_{i=1}^{a} (AB)_{ij}$$

$$\overline{B_j} = \frac{1}{na}B_j \quad (j = 1,2,\cdots,b)$$

$$T = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} X_{ijk} = \sum_{i=1}^{a}\sum_{j=1}^{b} (AB)_{ij}$$

$$\overline{X} = \frac{T}{N}$$

式中：$N=abn$ 是所有观测值的总数，$\overline{X}$是所有观察值的平均值。

利用上面所引入的符号，可以得到有交互作用的两因素方差分析的步骤如下：

1. 建立假设

由于两因素有交互影响，因此除了分别检验两因素单独对试验结果的影响外，还必须检验

两因素交互作用的影响是否显著。

对于因素 A：H_0：因素 A 的各个水平的影响无显著差异。

H_1：因素 A 的各个水平的影响有显著差异。

对于因素 B：H_0：因素 B 的各个水平的影响无显著差异。

H_1：因素 B 的各个水平的影响有显著差异。

对于因素 AB 的交互作用：

H_0：因素 AB 的各个水平的交互作用无显著影响。

H_1：因素 AB 的各个水平的交互作用有显著影响。

2. 离差平方和的分解

有交互作用的两因素方差分析的总离差平方和可以分解为 4 项：

$$\begin{aligned}\text{SST} &= \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(X_{ijk}-\overline{X})^2 = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}\{[X_{ijk}-(\overline{A}\,\overline{B})_{ij}]+[(\overline{A}\,\overline{B})_{ij}-\overline{A}_i-\overline{B}_j+\\ &\overline{X}]+(\overline{A}_i-\overline{X})+(\overline{B}_j-\overline{X})\}^2 = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(X_{ijk}-(\overline{A}\,\overline{B})_{ij})^2+n\sum_{i=1}^{a}\sum_{j=1}^{b}\\ &((\overline{A}\,\overline{B})_{ij}-\overline{A}_i-\overline{B}_j+\overline{X})^2+nb\cdot\sum_{i=1}^{a}(\overline{A}_i-\overline{X})^2+na\cdot\sum_{j=1}^{b}(\overline{B}_j-\overline{X})^2\end{aligned}$$

总离差平方和 SST 的自由度为 $N-1$。

分别记

$$\text{SSA} = nb\cdot\sum_{i=1}^{a}(\overline{A}_i-\overline{X})^2$$

为因素 A 的离差平方和，自由度为 $a-1$。

$$\text{SSB} = na\cdot\sum_{j=1}^{b}(\overline{B}_j-\overline{X})^2$$

为因素 B 的离差平方和，自由度为 $b-1$。

$$\text{SSE} = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(X_{ijk}-(\overline{A}\,\overline{B})_{ij})^2$$

表示随机误差的离差平方和，自由度为 $N-ab=abn-ab=ab(n-1)$。

$$\text{SSAB} = n\sum_{i=1}^{a}\sum_{j=1}^{b}((\overline{A}\,\overline{B})_{ij}-\overline{A}_i-\overline{B}_j+\overline{X})^2$$

表示因素间交互作用的离差平方和，自由度为

$$(N-1)-(a-1)-(b-1)-(n-1)ab=(a-1)(b-1)$$

3. 编制方差分析表，进行 F 检验

从方差分解式所得到的 SSA，SSB，SSAB 和 SSE 除以各自的自由度，就得到各自相应的

均方差，然后对因素A、因素B和因素AB的交互作用分别作F检验。与前面所讨论的情形一样，双因素方差分析表如表5.5所列。

表5.5 有交互影响的双因素方差分析表

方差来源	离差平方和	自由度	均方差	统计检验量 F
因素A	SSA	$a-1$	$\mathrm{MSA}=\dfrac{\mathrm{SSA}}{a-1}$	$F_A=\dfrac{\mathrm{MSA}}{\mathrm{MSE}}$
因素B	SSB	$b-1$	$\mathrm{MSB}=\dfrac{\mathrm{SSB}}{b-1}$	$F_B=\dfrac{\mathrm{MSB}}{\mathrm{MSE}}$
交互作用	SSAB	$(a-1)(b-1)$	$\mathrm{MSAB}=\dfrac{\mathrm{SSAB}}{(a-1)(b-1)}$	$F_{AB}=\dfrac{\mathrm{MSAB}}{\mathrm{MSE}}$
误差E	SSE	$n-ab$	$\mathrm{MSE}=\dfrac{\mathrm{SSE}}{N-ab}$	
总方差	SST	$n-1$		

与前面所讨论过的一样，根据方差分析表计算得到 F_A、F_B 和 F_{AB} 以后，根据问题的显著性水平 α，查表分别得到 $F_\alpha\{(a-1),(n-ab)\}$，$F_\alpha\{(b-1),(n-ab)\}$ 和 $F_\alpha\{(a-1)(b-1),(n-ab)\}$。于是可以分别检验因素A和B的影响，以及两因素的交互作用的影响是否显著。

对于因素A而言，若 $F_a>F_\alpha\{(a-1),(n-ab)\}$，就拒绝关于因素A的原假设，说明因素A对结果有显著的影响；否则，就接受原假设，说明因素A对结果没有显著的影响。对于因素B而言，若 $F_B>F_\alpha\{(b-1),(n-ab)\}$，则拒绝关于因素B的原假设，说明因素B对结果有显著的影响；否则，就接受原假设，说明因素B对结果没有显著的影响。对于两因素的交互作用，若 $F_{AB}>F_\alpha\{(a-1)(b-1),(n-ab)\}$，则拒绝关于两因素交互作用的原假设，说明因素A和因素B对结果有显著交互影响；否则，就接受原假设，说明两因素对结果没有显著的交互影响。

5.7.5 双因素方差分析函数

对于一个双因素模型，分别用A，B表示。其因子模型如下：

$$y_{ijk}=\mu+A_{.j}+B_{j.}+(AB)_{ij}+\epsilon_{ijk}$$

式中：AB 表示A，B共同影响。

调用方式

```
p = anova2(X, reps)
p = anova2(X,reps,'displayopt')
[p,table] = anova2(X, reps)
[p,table,stats] = anova2(X, reps)
```

输入参数

X	样本观察值
reps	每个单元中观察量的个数

输出参数

p	零假设成立的概率
table	方差分析表
stats	样本方差分析统计信息

5.7.6　多因素方差分析函数

下面以三因素模型为例说明多因素模型，三因素分别用 A、B、C 表示，三因素模型如下：

$$y_{ijkl}=\mu+A_{.j.}+B_{i..}+C_{..k}+(AB)_{ij.}+(AC)_{i.k}+(BC)_{.jk}+(ABC)_{ijk}+\epsilon_{ijkl}$$

式中：AB 是 A、B 交互因子；BC 是 B、C 交互因子；AC 是 A、C 交互因子；ABC 是 A、B、C 共同影响；y_{ijkl} 是样本值，μ 是参数。

调用方式

```
p = anovan(x,group)
p = anovan(x,group,'Param1',val1,'Param2',val2,...)
[p,table,stats] = anovan(...)
[p,table,stats,terms] = anovan(...)
p = anovan(x,group,'sstype', integer)
```

输入参数

X	样本值，每列为同一变量
group	观察值所在的类别
Param1	参数1的名称
val1	参数1的值，参数名称及其取值如表5.6所列。

表 5.6　多因子分析参数名称及其取值

参数名称	参数值
'sstype'	表示方差求和的方式，取值为1,2,3(默认值)，sstype的取值只影响非平衡数据的计算
'varnames'	因子名称，默认值为'X_1'、'X_2'…
'display'	'on'表示显示方差表、盒图，'off'不显示
'random'	随机效应
'alpha'	置信度

续表 5.6

参数名称	参 数 值
'model'	可以为'Linear'、'interaction'、'full'、整数、矩阵。linear(默认值)仅计算 N 个主效应,interaction 计算 N 个主效应和二因子交互效应,full 计算 N 个主效应和所有交互水平,整数 k 计算第 k 个水平交互作用,矩阵为主效应和交互作用

'model'如果取值为向量或矩阵的含义如表 5.7 所列。

表 5.7 三因子模型 model 取值为向量时的含义

行向量	含 义
[1 0 0]	仅考虑因子 A 的影响
[0 1 0]	仅考虑因子 B 的影响
[0 1 0]	仅考虑因子 C 的影响
[1 1 0]	考虑因子 A、B 的交互影响
[0 1 1]	考虑因子 B、C 的交互影响
[1 0 1]	考虑因子 A、C 的交互影响
[1 1 1]	考虑因子 A、B、C 的交互影响

如果 model= $\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}$ 时表示考虑 B 因子、C 因子、BC(B,C 交互影响)的影响。

输出参数

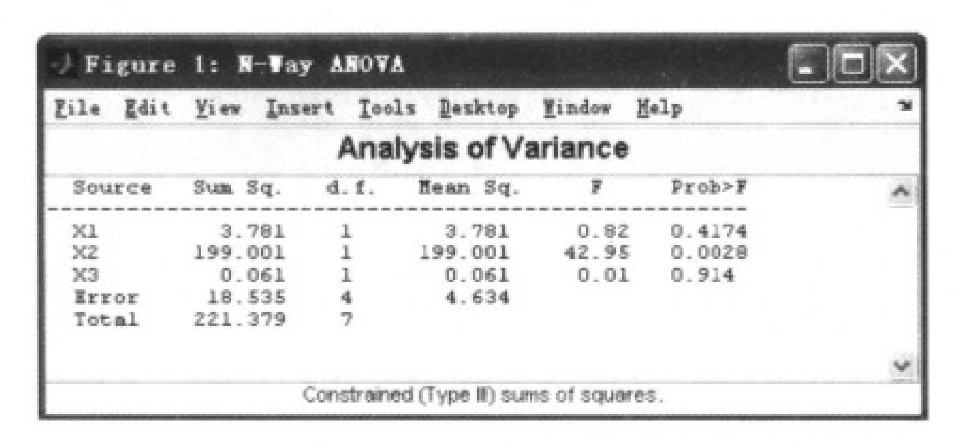
```
p          各因子对应的概率值
table      方差分析表
stats      模型特征,各统计指标
terms      model 的内容
```

【例 5-6】 下面是一个三因子方差分析的例子,y 是观察值,g_1,g_2,g_3 代表不同的因子。

```
>> y = [52.7 57.5 45.9 44.5 53.0 57.0 45.9 44.0]';
>> g1 = [1 2 1 2 1 2 1 2];
>> g2 = {'hi';'hi';'lo';'lo';'hi';'hi';'lo';'lo'};
>> g3 = {'may'; 'may'; 'may'; 'may'; 'june'; 'june'; 'june'; 'june'};
>> p = anovan(y, {g1 g2 g3})              % 不考虑因子之间交互影响
p =
    0.4174
    0.0028
```

```
0.9140
```

三因子方差分析表如图5.16所示。


Figure 1: N-Way ANOVA

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	3.781	1	3.781	0.82	0.4174
X2	199.001	1	199.001	42.95	0.0028
X3	0.061	1	0.061	0.01	0.914
Error	18.535	4	4.634		
Total	221.379	7			

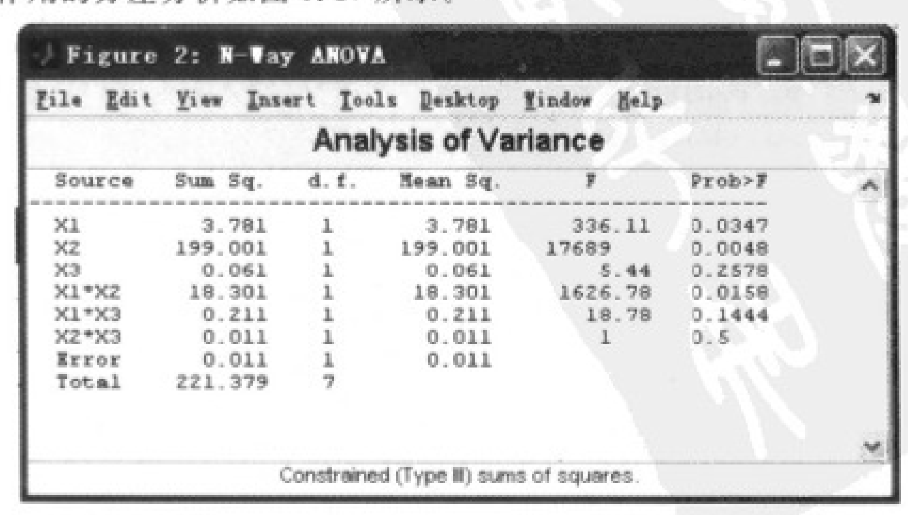
Constrained (Type III) sums of squares.

图5.16 三因子方差分析图

如果考虑各个因素之间的交互影响，就需要选用'model'的值为'interaction'。

```
>> p = anovan(y, {g1 g2 g3}, 'model', 'interaction') % 考虑因子之间两两交互影响
p =
    0.0347
    0.0048
    0.2578
    0.0158
    0.1444
    0.5000
```

含有交互作用的方差分析如图5.17所示。


Figure 2: N-Way ANOVA

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	3.781	1	3.781	336.11	0.0347
X2	199.001	1	199.001	17689	0.0048
X3	0.061	1	0.061	5.44	0.2578
X1*X2	18.301	1	18.301	1626.78	0.0158
X1*X3	0.211	1	0.211	18.78	0.1444
X2*X3	0.011	1	0.011	1	0.5
Error	0.011	1	0.011		
Total	221.379	7			

Constrained (Type III) sums of squares.

图5.17 含有交互作用的方差分析图

如果仅考虑B,C,BC因子的影响时可以执行下面命令。

```
>> [p table stats terms] = anovan(y, {g1 g2 g3}, 'model', [0 1 0;0 0 1;0 1 1])
p =
    0.0039
    0.9216
    0.9663
table =
    'Source'    'Sum Sq.'    'd.f.'    'Singular?'    'Mean Sq.'    'F'         'Prob>F'
    'X2'        [199.0013]   [1]       [0]            [199.0013]    [35.6873]   [0.0039]
    'X3'        [0.0613]     [1]       [0]            [0.0613]      [0.0110]    [0.9216]
    'X2 * X3'   [0.0113]     [1]       [0]            [0.0113]      [0.0020]    [0.9663]
    'Error'     [22.3050]    [4]       [0]            [5.5762]      []          []
    'Total'     [221.3788]   [7]       [0]            []            []          []
stats =
        source: 'anovan'
         resid: [8x1 double]
        coeffs: [9x1 double]
           Rtr: [4x4 double]
      rowbasis: [4x9 double]
           dfe: 4
           mse: 5.5762
   nullproject: [9x4 double]
         terms: [3x3 double]
       nlevels: [3x1 double]
      termcols: [4x1 double]
    coeffnames: {9x1 cell}
          vars: [9x3 double]
      varnames: {3x1 cell}
      grpnames: {3x1 cell}
           ems: []
         denom: []
       dfdenom: []
       msdenom: []
        varest: []
         varci: []
      txtdenom: []
        txtems: []
```

```
        rtnames: []
terms =
        0     1     0
        0     0     1
        0     1     1
```

B,C,BC 因子方差分析表如图 5.18 所示。

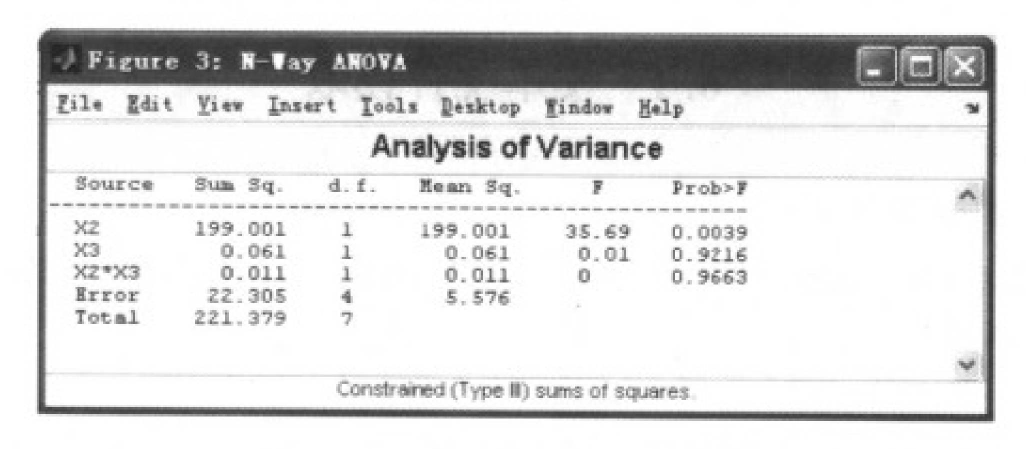


Figure 3: N-Way ANOVA

File Edit View Insert Tools Desktop Window Help

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X2	199.001	1	199.001	35.69	0.0039
X3	0.061	1	0.061	0.01	0.9216
X2*X3	0.011	1	0.011	0	0.9663
Error	22.305	4	5.576		
Total	221.379	7			

Constrained (Type III) sums of squares.

图 5.18　B,C,BC 因子方差分析表

思考题

1. 生成 10 个均值为 0、方差为 1 的正态分布随机数，然后求其偏度、峰度。

2. 比较 A 股三只不同行业股票的收益率，利用因子模型观察分析是否存在行业间差异。

3. 计算深发展(000001. SZ)2006 年的日收益率的均值、方差、偏度及峰度，建立 CAPM 模型。

第6章　数据文件读取和金融数据处理

数据文件的读取是数据处理的基本功，本章讨论了不同格式数据文件的读取，读者应掌握文本数据、Excel 格式数据的读写，掌握时间序列格式数据之间的转换。

6.1　文本文件读取

6.1.1　读取目录内容

MATLAB 的 dir 函数负责读入文件夹中的文件，例如读入根目录下扩展名为.txt 文件可以执行下面命令。

```
>> file=dir('C:\*.txt')
```

目录中扩展名为.txt 文件的特征保存在结构变量 file 中，file.name 保存了所有文件的名称。

```
>> file.name
ans =
      ckinfo.txt
ans =
      asd.txt
ans =
      dir.txt
ans =
      test1.txt
ans =
      test2.txt
ans =
      test3.txt
```

6.1.2　fprintf 函数写入数据

有时需要将数据内容写入文本文件中，函数 fprintf 可以将数据显示在显示屏上或者文件中。

调用方式

```
count = fprintf(fid, format, A, …)
```

输入参数

fid　　是一个整数型变量，它可以是 fopen 命令成功返回的文件标识符，也可以是 MATLAB 保留的具有特殊意义的数值。1 表示标准输出，也就是屏幕输出；2 表示标准错误输出，同样也是屏幕输出

format　　输出格式控制，变量 format 是一个字符串，用法类似 C 语言格式，主要控制符号、排序、重要数字及字符宽度等。例如 % -12.5e，其中 % 表示开始读人数据，- 表示标记，12 表示输入的宽度，5 表示小数位数，e 表示标记，输出格式内容如表 6.3 所列。format 以 % 开始，包含下面内容：

- 标志符 flags(可选项)，具体如表 6.1 所列
- 宽度和精度(可选项)，具体如表 6.2 所列
- 图表类型定义(可选项)
- 转换字符(必选项)，具体如表 6.3 所列

表 6.1　flags 内容

字　符	内　容	例　子	字　符	内　容	例　子
−	左对齐	%−5.2d	空格	在数字前加空格	% 5.2d
+	显示符号	%+5.2d	0	使用 0 补齐	%05.2d

表 6.2　宽度和精度内容

记　号	内　容	例　子
Field 宽度	一个数字字符串，定义被打印的数字最小数目	% 6f
Precision 精度	一个数字字符串，包含一个句点(.)，定义小数点右边位数	%6.2f

表 6.3　转换符内容

记　号	内　容	记　号	内　容
%c	单个字符	%g	文件中的%e，或者%f 更简洁形式，不重要的 0 不打印
%d	小数记号		
%e	指数形式，如 3.1415e+00	%G	同上。只是 e 换成 E
%E	指数形式，如 3.1415E+00	%i	十进位
%f	固定点符号	%o	八进位

续表 6.3

记　号	内　容	记　号	内　容
%s	写入字符串	%x	十六进位(小写)
%u	十进位(不带符号)	%X	十六进位(大写)

表 6.4 是换行符、换页符的内容。

表 6.4　换行符、换页符的内容

字　符	内　容	字　符	内　容
\b	后退键	\t	水平制表符
\f	进纸	\\	表示反斜杠
\n	换行符	\' or '	单引号
\r	回车键	%%	一个百分号

A　　　　　　　　MATLAB 中的变量

输出参数

count　　　　　　表示成功写入文件的字节数

例如要在屏幕上显示“It's Friday.”只需输入

```
>> fprintf(1,'It''s Friday.\n')
It's Friday.
```

如果在字符串中夹带显示变量的内容,则可以用“%g”进行提示。

```
>> fprintf('A unit circle has circumference %g radians.\n',2 * 2.5)
A unit circle has circumference 5 radians.
```

如果变量是矩阵,则每列分别输出到屏幕上。

```
>> B = [8.8   7.7; 8800   7700];
>> fprintf(1, 'X 为 %6.2f 米或者 %8.3f 米\n', 9.9, 9900, B)
```

显示内容如下:

```
X 为   9.90 米或者 9900.000 米
X 为   8.80 米或者 8800.000 米
X 为   7.70 米或者 7700.000 米
>>  fprintf('圆的周长是 %g 米.\n',2 * 2.5)
```

显示内容如下：

```
圆的周长是5米.
```

如果需要输入满足约束条件的矩阵时，则可以输入下面命令：

```
>> a=[1 2 ;3 4];
>> fprintf('%2.2f    %2.2f\n',a(a>2))
3.00   4.00
```

这样矩阵 **a** 中大于2的元素显示在屏幕上。

例如把变量内容输入到文件中：

```
>> x = 0:.1:1;
>> y = [x; exp(x)];
>> fid = fopen('exp.txt', 'wt');            %'wt'表示写入数据操作
>> fprintf(fid, '%6.2f %12.8f\n', y);
>> fclose(fid)
```

查看文件 exp.txt 内容如下：

```
>> ! type exp.txt
  0.00    1.00000000
  0.10    1.10517092
  0.20    1.22140276
  0.30    1.34985881
  0.40    1.49182470
  0.50    1.64872127
  0.60    1.82211880
  0.70    2.01375271
  0.80    2.22554093
  0.90    2.45960311
  1.00    2.71828183
```

6.1.3　fscanf 函数读出数据

调用方式

```
A = fscanf(fid, format)
[A,count] = fscanf(fid, format, size)
```

输入参数

fid	同 6.1.2 节所述
format	同 6.1.2 节所述
size	决定读取多少数据。其内容如下： n 表示将 n 个元素读入一个向量； inf 读取到文件结尾，结果是一个向量； [m,n]读取的元素填满 m 行 n 列矩阵

输出参数

A	数据变量
count	读取值的个数

下面读出 exp.txt 文件。

```
>> fid = fopen('exp.txt', 'r');                    % 'r'表示读取文件操作
>> a = fscanf(fid, '%g %g', [2 inf]);
>> fclose(fid)
>> a = a'
>> a
a =
     0          1.0000
     0.1000     1.1052
     0.2000     1.2214
     0.3000     1.3499
     0.4000     1.4918
     0.5000     1.6487
     0.6000     1.8221
     0.7000     2.0138
     0.8000     2.2255
     0.9000     2.4596
     1.0000     2.7183
```

6.1.4 从文本文件中读入格式化数据

1. textread 函数读取文本文件

MATLAB 从文本文件中读取格式化的数据函数是 textread。

调用方式

```
[A,B,C,...] = textread('filename','format')
[A,B,C,...] = textread('filename','format',N)
[...] = textread(...,'param','value',...)
```

输入参数

'filename'	需要读取的数据文件
'format'	每行各个数据格式
N	读取数据时使用 N 次

当调用 textread 函数时会按照指定的格式从'filename'中读取数据，并将数据分别保存在变量 A，B，C 中，直到文件内容被读完为止。

下面是读取 c:\中的 test3.txt 文件操作。

```
>> ! type c:\test3.txt
Q1 1 2
Q2 3 4
>> [a1 a2 a3] = textread('c:\test3.txt','%s %f %f')
a1 =
     'Q1'
     'Q2'
a2 =
     1
     3
a3 =
     2
     4
```

这样文件 test3.txt 中的每行分别读入变量 a1，a2，a3。

2. textscan 函数读取文本数据

对于具有多行文字说明文本数据文件，最好的办法则是用 textscan 函数读取，或者是用 fscanf 函数读取。

调用方式

```
C = textscan(fid, 'format')
C = textscan(fid, 'format', N)
C = textscan(fid, 'format', param, value, ...)
C= textscan(fid, 'format', N, param, value, ...)
```

输入参数

fid　　为 fopen 命令返回的文件标识符

format　　是一个字符串变量，表示读取数据及数据转换的规则。format 中包含了多种方式，其内容如表 6.5 所列

表 6.5　format 内容

数据转换规则	说　明
%n	读取一个数字并转换为 double
%d	读取一个数据并转换为 int32
%d8	读取一个数据并转换为 int8
%d16	读取一个数据并转换为 int16
%d32	读取一个数据并转换为 int32
%d64	读取一个数据并转换为 int64
%u	读取一个数据并转换为 uint32
%u8	读取一个数据并转换为 uint8
%u16	读取一个数据并转换为 uint16
%u32	读取一个数据并转换为 uint32
%u64	读取一个数据并转换为 uint64
%f4	读取一个数字并转换为 double
%f32	读取一个数字并转换为 float
%f64	读取一个数字并转换为 double
%s	读取一个字符串
%q	读取一个可以是双引号括着的字符串
%c	读取一个字符，包括空格
%[…]	读取和括号中字符串相匹配的字符。读取操作在首次遇到不匹配的字符和空格时停止。例如%[mus]把 summer 读成 summ
%[^…]	读取和括号中字符串不匹配的字符。读取操作在首次遇到不匹配的字符和空格时停止。例如%[^mus]把 summer 读成 er

textscan 所支持的数字单元宽度声明有两种，具体如表 6.6 所列。

表 6.6　textscan 支持的数据读入格式

名　称	说　明
N	读取包括小数点在内的 *N* 个数字或者遇到的分隔符，例如%5f32 会把 473.238 读为 473.2
N.D	读取包括小数点在内的 *N* 个数字或者遇到的分隔符，返回的数字在小数点后有 *D* 位数，例如%7.2f 会把 473.238 读为 473.23

下面建立新文件 test1. txt，文件前两行是字符串，后三行是数组，首先查看其内容。

```
>> ! type  c:\test1.txt
a1
a2
1
2
```

注意，在 Command 窗口下输入"！"后就可以直接执行 DOS 命令。

```
>> file = fopen('c:\test1.txt','r');                    % 打开需要读取的文件
>> str1 = textscan(file,'%s',2) ;
>> str1{1}
ans =
      'a1'
      'a2
```

如果需要读取第三行、第四行可以执行下面命令。

```
>> da = textscan(file,'%f',[3 4]) ;
>> da{1}
ans =
      1
      2
```

由于 da 是单元数据，下面将其转换为实数型。

```
>> cell2mat(da)
ans =
      1
      2
```

如果数据文件中有间隔符，则需要忽略间隔符。

```
>> type c:\test3.txt
1    , 1   , 2
32   , 3   , 4
>> tt = fopen('c:\test3.txt');
>> var = textscan(tt,'%f   %f %f','delimiter',',')          % 忽略','
>> aa = cell2mat(var)
aa =
      1       1       2
```

```
      32       3       4
```

结束时需要关闭文件。

```
>> fclose(file)
>> ! type c:\test2.txt
 2 5 3 6
 2 3 5
 1 9 5 4 6
>> fid = fopen('c:\test2.txt');
>> for i = 1:3 ,a{i} = fgets(fid); end
>> fclose(fid);
```

6.1.5 带有间隔符的文本数据读写

1. 读入带有间隔符的数据文件

有时数据与数据之间带有间隔符，需要跳过这些间隔符，才能读入下一个数据，MATLAB 的 dlmread 函数可以执行此项功能。注意，该函数只能读入数据文件，不能读入非数据文件。

调用方式

```
M = dlmread('filename')
M = dlmread('filename', delimiter)
M = dlmread('filename', delimiter, R, C)
M = dlmread('filename', delimiter, range)
```

输入参数

'filename'	等待读入的数据文件
delimiter	数据文件中有间隔符，默认的间隔符是逗号
R、C	读入数据的起始位置
rang	读入数据的位置，前面两个表示起始的行与列，后面表示数据区域的终止行与列

注意，如果数据文件没有间隔符，delimiter 可以用' '代替。

例如 myfile.txt 文件是 3 阶的魔方阵。

```
>> ! type myfile.txt
 8,1,6
 3,5,7
 4,9,2
```

下面读入 MATLAB 的 ***M*** 矩阵。

```
>> M = dlmread('myfile.txt', ',')
M =
     8     1     6
     3     5     7
     4     9     2
```

如果仅仅读入第 3 行，则可以执行如下命令：

```
>> M = dlmread('myfile.txt', ',',2,0)
M =
  4     9     2
```

注意数据起始于第 0 行第 0 列。

2. 写入带有间隔符的数据文件

调用方式

```
dlmwrite('filename', M)
dlmwrite('filename', M, 'D')
dlmwrite('filename', M, 'D', R, C)
dlmwrite('filename', M, attribute1, value1, attribute2, value2, ...)
dlmwrite('filename', M, '-append')
dlmwrite('filename', M, '-append', attribute-value list)
```

输入参数

filename	写入数据的文件名
M	MATLAB 中变量
'D'	间隔符
R	数据起始的行
C	数据起始的列，$R=0$，$C=0$ 表示从第 1 行第 1 列开始写入数据
'-append'	添加数据操作，否则将删除旧数据
attribute1	属性 1
value1	属性 1 的值。其内容如表 6.7 所列

表 6.7　dlmwrite 函数属性的内容

属　性	内　容	属　性	内　容
delimiter	间隔符	coffset	新老数据间隔列数
roffset	新老数据间隔行数	precision	数据精度，如'%10.5f'

下面将3阶魔方矩阵写入文件myfile.txt中：

```
>> M = magic(3);
>> dlmwrite('myfile.txt', M, 'delimiter', '\t', 'precision', 6)
>> ! type  myfile.txt
    8     1     6
    3     5     7
    4     9     2
```

如果需要空一行后再添加新数据，则可以输入如下命令：

```
>> dlmwrite('myfile.txt', M, 'delimiter', '\t','roffset',1,'-append')
>> ! type myfile.txt
    8     1     6
    3     5     7
    4     9     2
    8     1     6
    3     5     7
    4     9     2
```

如果保留2位小数，则输入如下命令：

```
>> dlmwrite('myfile.txt', M, 'delimiter', ',', 'precision', '%1.2f');
>> type myfile.txt
    8.00,1.00,6.00
    3.00,5.00,7.00
    4.00,9.00,2.00
```

6.1.6 Excel数据文件读写

1. 读出Excel文件数据

调用方式

```
num = xlsread('filename')
```

如果filename中有非双精度数据则会忽略该数据，会导致数据减少。

```
num = xlsread('filename', -1)
num = xlsread('filename', sheet)
num = xlsread('filename', 'range')
num = xlsread('filename', sheet, 'range')
```

```
[num, txt] = xlsread('filename', sheet, 'range',)
```

输入参数

'filename'	Excel 格式的数据文件名称
sheet	Excel 文件中的表的名称
'range'	表中单元格的区域

输出参数

num	读入 Excel 文件数据
txt	保存文本内容

下面把厦门建发 2005 年 9 月 19 日至 29 日的收盘价、成交量数据存入 Excel 中，命名为 book1.xls，并从 MATLAB 中将其读出来；因为文件中有日期型数据，这样可以将数据分成两个部分，xmjf_date 为单元数据储存日期，xmjf_600153 储存收盘价、成交量。执行如下命令：

```
>> [xmjf_600153,xmjf_date] = xlsread('book1.xls');
```

图 6.1 为 MATLAB 浏览数据的结果。

Array Editor

xmjf_600153

	1	2	3	4
1	4.07	21578.7		
2	4	29130.2		
3	3.92	22623.7		
4	3.84	34239.7		
5	3.8	12593.9		
6	3.84	10715.1		
7	3.75	12640.1		

xmjf_date

	1	2	3	4
1	'2005-9-19'			
2	'2005-9-20'			
3	'2005-9-21'			
4	'2005-9-22'			
5	'2005-9-23'			
6	'2005-9-26'			
7	'2005-9-27'			
8	'2005-9-28'			
9	'2005-9-29'			

xmjf_600153　xmjf_date

图 6.1　厦门建发股价

例如，Excel 中的 testdata1. xls 文件内容如下：

```
1	6
2	7
3	8
4	9
5	10
```

现将该文件读入 MATLAB 中，数据保存在变量 A 中。

```
>> A = xlsread('testdata1.xls')
A =
     1     6
     2     7
     3     8
     4     9
     5    10
```

如果 Excel 中出现文字，则将该单元格的数据读为 NaN。例如 Excel 中的 testdata1. xls 文件内容如下：

```
1	6
2	7
3	8
4	9
5	Read
```

读入文件如下：

```
>> A = xlsread('testdata1.xls')
A =
     1     6
     2     7
     3     8
     4     9
     5   NaN
```

当 Excel 文件中含有多个表时，需要具体指定第几个表，给出单元格的起始位置。例如读取文件 testdata2. xls 的第一个表，数据区域起始于 A4 止于 B5。

```
>> A = xlsread('testdata2.xls',1, 'A4:B5');
A =
```

```
4     9
5     NaN
12    98
13    99
14    97
```

如果第一个表的名称为 Temperatures，也可以指明，然后选择具体的数据起始位置。

```
>> ndata = xlsread('tempdata.xls', 'Temperatures')
ndata =
        12     98
        13     99
        14     97
```

如果 Excel 文件中有文件头，则需要增加一个输出变量保存数据头。例如 tempdata.xls 的内容如下：

```
Time    Temp
12      98
13      99
14      97
>> [ndata, headertext] = xlsread('tempdata.xls', 'Temperatures')
ndata =
        12     98
        13     99
        14     97
headertext =
            'Time'     'Temp'
```

2. 写入 Excel 数据文件

通常也可以将 MATLAB 中的数据读入 Excel 中，函数是 xlswrite。

调用方式

```
xlswrite('filename', M)
xlswrite('filename', M, sheet)
xlswrite('filename', M, 'range')
xlswrite('filename', M, sheet, 'range')
```

输入参数

filename　　　　Excel 文件名

```
M                    MATLAB 中变量
sheet                Excel 中工作簿
range                Excel 中工作簿的数据区域.
```

下面在 MATLAB 中创建一个魔方矩阵，然后将其写入 Excel 中。

```
>> M = magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
>> % 将数据写入 Excel 文件 abc 中工作簿 Sheet1 中，位置从 A2 到 C4。
>> xlswrite('abc',M,'Sheet1','A2:C4')          % 将数据写入
```

打开 abc. xls，可以看到数据已经写入文件 abc. xls 中。

6.2 创立时间序列变量

6.2.1 时间序列数组的创立和数据文件读取

由于金融数据大部分表现为时间序列，为了便于运算与绘图，MATLAB 有专门的时间序列格式保存时间序列数据。时间序列变量的扩展名为. fints，该变量把时间数据保存在第一列，其他列为观察值，时间序列变量运算时变量的内容发生变化，但时间不变。下面介绍 MATLAB 中时间变量的处理函数。

1. 利用 fints 函数创立日期型数组

在 MATLAB 中创立日期型数组的函数是 fints。例如

```
>> price  = [1:6]'
price =
       1
       2
       3
       4
       5
       6
>> dates = [today:today + 5]'
```

```
dates =
        732721
        732722
        732723
        732724
        732725
        732726
>> tsobjkt = fints(dates, price)
tsobjkt =
        desc:  (none)
        freq:  Unknown (0)
        'dates:  (6)'      'series1:  (6)'
        '13 - Feb - 2006'    [              1]
        '14 - Feb - 2006'    [              2]
        '15 - Feb - 2006'    [              3]
        '16 - Feb - 2006'    [              4]
        '17 - Feb - 2006'    [              5]
        '18 - Feb - 2006'    [              6]
```

利用 whos 命令查看内存中变量，显示变量的信息如下：

```
>> whos
  Name          Size                       Bytes     Class
  price         6x1                        48        double array
  dates         6x1                        48        double array
  tsobjkt       6x1                        1266      fints object
Grand total is 66 elements using 1362 bytes
```

从输出结果中看到变量 tsobjkt 是 fints 型数组。

2. 金融时间序列文件读取

MATLAB 中的 ascii2fts 函数把 Ascii 文件内容保存为 fints 型时间序列变量。
调用方式

```
tsobjkt = ascii2fts(filename, timedata, descrow, colheadrow, skiprows)
```

输入参数

filename	文件名，用单引号括起来
timedata	判定是不是"天"数据，如果是，输入字符串't'，不是则输入'nt'

```
descrow                  确定 Ascii 文件中文字说明的行数
colheadrow               说明每列变量名所在的行数
skiprows                 Ascii 文件中不需要输入的行
```

输出参数

```
tsobjkt                  MATLAB 中 fints 型时间序列数据
```

下面用 DOS 中的 type 命令显示 work 目录下的 at.txt 文件内容。

```
>> ! type at.txt
    16 - Feb - 2006      1
    17 - Feb - 2006      2
    18 - Feb - 2006      3
    19 - Feb - 2006      4
    20 - Feb - 2006      5
    21 - Feb - 2006      6
```

下面用 DOS 中的 rename 命令改变文件的名称。

```
>> ! rename at.txt at.dat             % 利用 DOS 命令将.txt 格式文件命名为.dat 文件
>> tsobjkt = ascii2fts('at.dat')      % 读取 at.dat 文件中的数据
```

这样 at.dat 中的数据读入 MATLAB 中，变量名为 tsobjkt，类型为 fints 型数据，打开 tsobjkt 就可以看见具体内容。代码如下：

```
tsobjkt =
     desc:   at.dat
    freq:   Unknown (0)
    'dates:  (6)'     'series1:  (6)'
    '16 - Feb - 2006'     [             1]
    '17 - Feb - 2006'     [             2]
    '18 - Feb - 2006'     [             3]
    '19 - Feb - 2006'     [             4]
    '20 - Feb - 2006'     [             5]
    '21 - Feb - 2006'     [             6]
```

下面观察文件 at.dat 的内容。

```
>> ! type at.dat
ABC Company Stock
16 - Feb - 2006      1
17 - Feb - 2006      2
```

```
18 - Feb - 2006      3
19 - Feb - 2006      4
20 - Feb - 2006      5
21 - Feb - 2006      6
```

由于第一行不是数据而是描述数据的内容，因此直接调用函数 ascii2fits 就无法识别该文件。

```
>> tsobjkt = ascii2fts('at.dat')
```

显示错误信息如下：

```
?? Error using = = > ascii2fts
Either the file contents do not correspond to the input
information, or the file contents contain time information
which was not indicated in the function call.
The actual error generated when reading the text file via
TEXTREAD is:
Trouble reading floating point number from file (row 1, field 2) = = > open
```

由于 at.dat 的第一行不是数字，ascii2fts 函数无法辨认第一行，这时需要把第一行作为描述内容读取，执行如下命令：

```
>> tsobjkt = ascii2fts('at.dat',1)
```

这样 at.dat 的第一行就作为描述性内容保存到 tsobjkt 中，其内容如下：

```
tsobjkt =
     desc:  ABC Company Stock
    freq:  Unknown (0)
    'dates:  (6)'     'series1:  (6)'
    '16 - Feb - 2006'    [            1]
    '17 - Feb - 2006'    [            2]
    '18 - Feb - 2006'    [            3]
    '19 - Feb - 2006'    [            4]
    '20 - Feb - 2006'    [            5]
    '21 - Feb - 2006'    [            6]
```

进一步地，如果 at.dat 的内容如下：

```
>> ! type at.dat
ABC Company Stock
DATE          CLOSE
16 - Feb - 2006      1
```

```
17 - Feb - 2006     2
18 - Feb - 2006     3
19 - Feb - 2006     4
20 - Feb - 2006     5
21 - Feb - 2006     6
```

at.dat文件中描述的内容有两行，第一行说明是ABC公司的股价，第二行说明是日期与收盘价。at.dat文件描述性内容的行数为两行，对数据的说明在第二行，执行下列命令：

```
>> tsobjkt = ascii2fts('at.dat',1,2)
```

这样at.dat的第一行就作为描述性内容保存到tsobjkt中。

```
tsobjkt =
          desc:  ABC stock prices
          freq:  Unknown (0)
          'dates:  (6)'      'CLOSE:  (6)'
          '16 - Feb - 2006'    [        1]
          '17 - Feb - 2006'    [        2]
          '18 - Feb - 2006'    [        3]
          '19 - Feb - 2006'    [        4]
          '20 - Feb - 2006'    [        5]
          '21 - Feb - 2006'    [        6]
```

这样“close”内容也被读入tsobjkt中。

实际上assii2fts也可以直接读入.txt类型的文件，如可以执行下列命令：

```
>> tsobjkt = ascii2fts('at.txt')
tsobjkt =
    desc:  at.txt
    freq:  Unknown (0)
    'dates:  (6)'      'series1:  (6)'
    '16 - Feb - 2006'    [        1]
    '17 - Feb - 2006'    [        2]
    '18 - Feb - 2006'    [        3]
    '19 - Feb - 2006'    [        4]
    '20 - Feb - 2006'    [        5]
    '21 - Feb - 2006'    [        6]
```

如果不需要读入第4列、第5列(注意日期列不算)，则可以将这两列用中括号括出来[4,5]，例如ascii2fts('at.txt',[4,5])，这样tsobjkt中就不会把at.dat的第4列、第5列读进来。

```
>> ! type at.dat
Staples, Inc. (SPLS)
Daily prices
DATE     OPEN     HIGH     LOW      CLOSE    VOLUME
Starting date: 04/08/1996
Ending date:   04/07/1999
4/8/96   19.50    19.75    19.25    19.375   548500
4/9/96   19.75    20.125   19.375   20       1135900
>> tsobjkt = ascii2fts('at.dat', 1, 3, [4 5])
tsobjkt =
desc:  Staples, Inc. (SPLS)
freq:  Unknown (0)
'dates: (2)'          'OPEN: (2)'    'HIGH: (2)'    'LOW:  (2)'
'08-Apr-1996'         [19.5000]      [19.7500]      [19.2500]
'09-Apr-1996'         [19.7500]      [20.1250]      [19.3750]
```

如需要读入小时数据,则要规定读入文件的格式。

```
>> type myfts_file2.txt                    % 显示文件内容
My FTS with Time
dates      times      Data1      Data2
  01-Jan-2001      11:00      10.000000      5.000000
  01-Jan-2001      12:00      2.000000       0.000000
  02-Jan-2001      11:00      6.000000       8.000000
  02-Jan-2001      12:00      5.000000       4.000000
  03-Jan-2001      11:00      9.000000       6.000000
  03-Jan-2001      12:00      8.000000       8.000000
>> MyFts = ascii2fts('myfts_file2.txt','t',1,2,1)
MyFts =
    desc:  My FTS with Time
    freq:  Unknown (0)
    'dates:  (6)'        'times:  (6)'        'Data1:  (6)'        'Data2:  (6)'
    '01-Jan-2001'        '11:00'              [        10]         [         5]
    '     "     '        '12:00'              [         2]         [         0]
    '02-Jan-2001'        '11:00'              [         6]         [         8]
    '     "     '        '12:00'              [         5]         [         4]
    '03-Jan-2001'        '11:00'              [         9]         [         6]
    '     "     '        '12:00'              [         8]         [         8]
```

6.2.2 时间序列数组运算

1. 日期运算

1) 查找现在时刻

在 MATLAB 中查找现在时刻的函数是 now，表示从 0000 年 1 月 1 日至现在时刻的时间。

例如，现在时刻是 2007 年 1 月 22 日 17 点 44 分 49 秒，可以用 now 函数查找时间。

```
>> now
ans =
733064.7394612615
```

表示 0000 年 1 月 1 日至 2007 年 1 月 22 日 17 点 44 分 49 秒的时间是 733 064.739 461 261 5 天。

```
>> datestr(733064.7394612615)
ans =
      22 - Jan - 2007 17:44:49
```

2) 查询当天日期

Today 函数表示自 0000 年 1 月 1 日至当天的天数，如今天是 2006 年 2 月 6 日星期一，则用如下命令查找：

```
>> today
ans =
      732714
```

表示 0000 年 1 月 1 日至今有 732 714 天。

3) 序数型日期转换为字符串日期

datestr 函数与 today 函数的功能相反，如果知道距离 0000 年 1 月 1 日的天数，则用 datestr 函数就可以计算出日期，例如查询第 732 714 天的日期可以执行如下命令：

```
>> datestr(732714)
ans =
      06 - Feb - 2006
```

若需要控制输出格式就要增加控制参数。如果输出格式为月-天-年，小时：分钟：秒，每天分上午与下午（mmmm，dd，yyyy HH：MM：SS. FFF AM），则需要调用形式 dt = datestr(now，'mmmm dd，yyyy HH：MM：SS. FFF AM')，例如确定距离 0000 年 1 月 1 日为

733 064.739 461 261 5的时刻,可以执行如下命令:

```
>> datestr(733064.7394612615,'mmmm dd, yyyy HH:MM:SS.FFF AM')
ans =
      January 22, 2007   5:44:49.453 PM
```

对应的时刻为2007年1月22日下午5点44分49.453秒。如果写成符合我国习惯的日期型则可以输入下面命令:

```
>> datestr(733064.7394612615,'yyyy,mm, dd, HH:MM:SS.FFF AM')
ans =
      2007,01, 22,  5:44:49.453 PM
```

4) 确定每月的第几天

调用方式

```
Day = day(D)
```

输入参数

D　　　　如果是用整型数表示从0000年1月1日至今的天数,则也可以是日期型字符串

输出参数

Day　　　　月的天数

例如计算2007年1月22日的天数。代码如下:

```
>> day1 = day('22 - Jan - 2007')
day1 =
      22
```

2007年1月22日距离0000年1月1日的天数是733 064,同样可以确定天数。代码如下:

```
>> day1 = day(733064)
day1 =
      22
```

5) 查询星期

在MATLAB中查询星期是weekday函数。

调用方式

```
[N, S] = weekday(D)
```

输入参数

D　　如果是用整型数表示从 0000 年 1 月 1 日至今的天数，则也可以用日期型字符串来表示

输出参数

N　　星期

S　　星期的缩写

```
>> [n, s] = weekday(733064)
n =
    2
s =
   Mon
```

整数 733 064 表示的天数是 2007 年 1 月 22 日，n 表示是星期一，一周的第二天（按照西方的习惯，星期日是一周的第一天），s 是储存星期一英文缩写的变量。实际上如果直接给出 2007 年 1 月 22 日的日期格式也可以得到同样结果。

```
>> [n, s] = weekday('22 - Jan - 2007')
n =
    2
s =
   Mon
```

6）查询月份

MATLAB 用 month 函数查询月份。

调用方式

```
[MonthNum, MonthString] = month(D)
```

输入参数

D　　如果是用整型数表示从 0000 年 1 月 1 日至今的天数，则也可以用日期型字符串来表示

输出参数

MonthNum　　月份

MonthString　　月份的缩写

7）把字符串型日期转换为序数型日期

序数型日期表示距离 0000 年 1 月 1 日的天数，字符型日期转换为序数型日期的函数是 datenum。

调用方式

```
N = datenum(V)
N = datenum(S, F)
N = datenum(S, F, P)
N = datenum(S, P, F)
N = datenum(Y, M, D)
N = datenum(Y, M, D, H, MN, S)
N = datenum(S)
N = datenum(S, P)
```

输入参数

V	字符型日期,可以为年、月、日、小时或分
F	输出格式,'dd-mmm-yyyy'表示按照"天-月-年"格式输出
P	表示按照间隔P输出
M	月份
Y	年
D	天
H	小时
MN	分
S	日期型字符串,但 N = datenum(Y, M, D, H, MN, S)中的S表示秒

下面是一个例子。

```
>> datenum('03-aug-2000')
ans =
      730701
```

上面结果表示2000年8月3日距离0000年1月1日有730 701天。

例如,2003年10月24日12时45分7秒序数型日期表示如下:

```
>> datenum('24-Oct-2003 12:45:07')
ans =
      7.3188e+005
```

8) 将字符型日期以向量形式输出

调用方式

```
V = datevec(N)
V = datevec(S, F)
V = datevec(S, F, P)
[Y, M, D, H, MN, S] = datevec(...)
V = datevec(S)
```

```
V = datevec(S, P)
```

输入参数、输出参数同上。

9）计算日期间隔

计算两个日期之间的天数需要用 daysact 函数。例如，计算 2006 年 2 月 27 日至 2006 年 3 月 1 日之间的天数可以执行以下命令：

```
>> daysact('27-feb-2006','01-mar-2006')
ans =
     2
```

10）时间序列数据的合并

调用方式

```
newfts = merge(fts1, fts2, …, Method, RefObj)
```

输入参数

fts1、fts2	需要合并的时间序列
Method	合并方法，其内容如下： 'union' or 'u'（默认值）　返回所有元素； 'intersection' or 'i'　仅合并相同的元素； 'reftime' or 'r'　以 RefObj 的日期为基准进行合并
RefObj	基准日期，其值不作为合并的内容

输出参数

newfts	合并后的新数据

下面生成 3 个时间序列，然后将其合并：

```
>> dates = {'jan-01-2001'; 'jan-02-2001'; 'jan-03-2001'; 'jan-04-2001'; 'jan-06-2001'};
>> data = [1; 1; 1; 1; 1];
>> t1 = fints(dates, data);
>> dates = {'jan-02-2001'; 'jan-03-2001'; 'jan-04-2001'; 'jan-05-2001'};
>> data = [2; 2; 2; 2];
>> t2 = fints(dates, data);
>> dates = {'jan-03-2001'; 'jan-04-2001'; 'jan-05-2001'; 'jan-06-2001'};
>> data = [3; 3; 3; 3];
>> t3 = fints(dates, data);
>> t123 = merge(t1, t2, t3)
```

```
t123 =
    desc:   ||  ||
    freq:  Unknown (0)
    'dates: (6)'     'series1: (6)'
    '01 - Jan - 2001'    [              1]
    '02 - Jan - 2001'    [              1]
    '03 - Jan - 2001'    [              1]
    '04 - Jan - 2001'    [              1]
    '05 - Jan - 2001'    [              2]
    '06 - Jan - 2001'    [              1]
```

从结果可以看出合并的过程是这样的，首先以 $t1$ 为基准，按照从左向右的次序进行合并。注意，改变 $t1$、$t2$、$t3$ 的次序会改变合并结果。

11）特定日期抽取函数

在研究股市是否存在周末效应时就需要从日数据中抽取周、月、季度或半年度的数据。通常处理方法是很烦琐的，但 MATLAB 金融时间序列工具箱有专门的函数完成此项功能。

- todaily 函数：从时间序列中抽取日数据。
- toweekly 函数：从时间序列中抽取周末数据。
- tomonthly 函数：从时间序列中抽取月末数据。
- toquarterly 函数：从时间序列中抽取季度末数据。
- tosemi 函数：从时间序列中抽取半年度末数据。
- toannual 函数：从时间序列中抽取年度末数据。

【例 6-1】 下面以金牛股份(000937)为例，说明如何将分析家软件数据导入到 MATLAB 中，然后进行相关操作。分析家软件是一款比较好的股票行情分析软件，可以从分析家网站(http://www.fenxijia.com)免费下载。数据选取的时间是 2005 年 11 月 14 日至 2006 年 1 月 10 日。

步骤如下：

① 打开分析家软件，输入代码 000937 就出现金牛股份股价图，日 K 线窗口如图 6.2 所示。

② 单击鼠标右键，在弹出的快捷菜单中选择【复制数据】命令，弹出一个提示框，如图 6.3 所示。

③ 打开 Excel，将数据粘贴到 Excel 文件中，如图 6.4 所示。

④ 由于日期型格式是 yyyy/mm/dd，不符合 MATLAB 默认格式，单击"A"列，右击，选择【设置单元格格式】菜单项，就会出现图 6.5 所示的对话框。



图 6.2 金牛股份的日 K 线



图 6.3 金牛股份日交易数据复制图

⑤ 对日期型数据进行修改，选择如图 6.5 所示的格式，单击【确定】按钮，日期的显示格式发生了变化，如图 6.6 所示。

⑥ 拖动鼠标把所要复制的内容确定下来，右击，选择【复制】菜单项，打开 Windows 中记事本，把复制的内容粘贴到记事本中，注意把汉字翻译成英文，这样便于 MATLAB 识别，如图 6.7所示。

	A	B	C	D	E	F
1	'000937'	G 金　牛 日线				
2	日期	开盘	最高	最低	收盘	成交量
3	2005-12-14	4.88	4.93	4.81	4.88	67064
4	2005-12-15	4.9	5.01	4.88	4.89	106106
5	2005-12-16	4.89	4.94	4.84	4.94	52277
6	2005-12-19	4.93	4.98	4.87	4.88	57681
7	2005-12-20	4.87	5.01	4.86	4.99	96808
8	2005-12-21	5	5.07	4.95	4.97	83140
9	2005-12-22	4.96	5	4.93	4.98	38033
10	2005-12-23	4.99	5.09	4.98	5.08	89875
11	2005-12-26	5.09	5.14	5.06	5.07	82578
12	2005-12-27	5.09	5.09	4.98	5.02	63578
13	2005-12-28	5.02	5.05	4.94	4.97	52353
14	2005-12-29	4.97	5.12	4.95	5.11	99265
15	2005-12-30	5.11	5.13	5	5.03	67547
16	2006-1-4	5.03	5.1	5.02	5.08	62509
17	2006-1-5	5.09	5.16	5.05	5.15	105077
18	2006-1-6	5.15	5.29	5.12	5.22	116407
19	2006-1-9	5.22	5.31	5.2	5.28	88119
20	2006-1-10	5.28	5.3	5.15	5.25	80245
21	2006-1-11	5.25	5.25	5.1	5.12	99268

图 6.4　将分析家数据复制到 Excel 中



图 6.5　Excel 的日期设置

Book2

	A	B	C	D	E	F
1	'000937'	G 金 牛	日线			
2	日期	开盘	最高	最低	收盘	成交量
3	14-Dec-05	4.88	4.93	4.81	4.88	67064
4	15-Dec-05	4.9	5.01	4.88	4.89	106106
5	16-Dec-05	4.89	4.94	4.84	4.94	52277
6	19-Dec-05	4.93	4.98	4.87	4.88	57681
7	20-Dec-05	4.87	5.01	4.86	4.99	96808
8	21-Dec-05	5	5.07	4.95	4.97	83140
9	22-Dec-05	4.96	5	4.93	4.98	38033
10	23-Dec-05	4.99	5.09	4.98	5.08	89875
11	26-Dec-05	5.09	5.14	5.06	5.07	82578
12	27-Dec-05	5.09	5.09	4.98	5.02	63578
13	28-Dec-05	5.02	5.05	4.94	4.97	52353
14	29-Dec-05	4.97	5.12	4.95	5.11	99265
15	30-Dec-05	5.11	5.13	5	5.03	67547
16	4-Jan-06	5.03	5.1	5.02	5.08	62509
17	5-Jan-06	5.09	5.16	5.05	5.15	105077
18	6-Jan-06	5.15	5.29	5.12	5.22	116407
19	9-Jan-06	5.22	5.31	5.2	5.28	88119
20	10-Jan-06	5.28	5.3	5.15	5.25	80245
21	11-Jan-06	5.25	5.25	5.1	5.12	99268

Sheet1 Sheet2 Sheet3

图 6.6 改变日期型数据的显示方式

at - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

000937 jin niu gu fen

date	open	high	low	close	volum
14-Dec-05	4.88	4.93	4.81	4.88	67064
15-Dec-05	4.9	5.01	4.88	4.89	106106
16-Dec-05	4.89	4.94	4.84	4.94	52277
19-Dec-05	4.93	4.98	4.87	4.88	57681
20-Dec-05	4.87	5.01	4.86	4.99	96808
21-Dec-05	5	5.07	4.95	4.97	83140
22-Dec-05	4.96	5	4.93	4.98	38033
23-Dec-05	4.99	5.09	4.98	5.08	89875
26-Dec-05	5.09	5.14	5.06	5.07	82578
27-Dec-05	5.09	5.09	4.98	5.02	63578
28-Dec-05	5.02	5.05	4.94	4.97	52353
29-Dec-05	4.97	5.12	4.95	5.11	99265
30-Dec-05	5.11	5.13	5	5.03	67547
4-Jan-06	5.03	5.1	5.02	5.08	62509
5-Jan-06	5.09	5.16	5.05	5.15	105077
6-Jan-06	5.15	5.29	5.12	5.22	116407
9-Jan-06	5.22	5.31	5.2	5.28	88119
10-Jan-06	5.28	5.3	5.15	5.25	80245

图 6.7 粘贴数据到记事本

⑦ 将数据保存到 MATLAB 7/work 目录下的 at.txt 文件，用 type 命令显示内容如下：

```
>> ! type at.txt
000937    jin niu gu fen
          date          open     high     low      close     volum
          14-Dec-05     4.88     4.93     4.81     4.88      67064
          15-Dec-05     4.9      5.01     4.88     4.89      106106
          16-Dec-05     4.89     4.94     4.84     4.94      52277
          19-Dec-05     4.93     4.98     4.87     4.88      57681
          20-Dec-05     4.87     5.01     4.86     4.99      96808
          21-Dec-05     5        5.07     4.95     4.97      83140
          22-Dec-05     4.96     5        4.93     4.98      38033
          23-Dec-05     4.99     5.09     4.98     5.08      89875
          26-Dec-05     5.09     5.14     5.06     5.07      82578
          27-Dec-05     5.09     5.09     4.98     5.02      63578
          28-Dec-05     5.02     5.05     4.94     4.97      52353
          29-Dec-05     4.97     5.12     4.95     5.11      99265
          30-Dec-05     5.11     5.13     5        5.03      67547
          4-Jan-06      5.03     5.1      5.02     5.08      62509
          5-Jan-06      5.09     5.16     5.05     5.15      105077
          6-Jan-06      5.15     5.29     5.12     5.22      116407
          9-Jan-06      5.22     5.31     5.2      5.28      88119
          10-Jan-06     5.28     5.3      5.15     5.25      80245
```

下面用 ascii2fts 函数将 at.txt 中的数据读入 MATLAB 中，代码如下：

```
>> jinniu = ascii2fts('at.txt',2,2)
jinniu =
desc:   date        open     high     low      close    volum
freq:   Unknown (0)
'dates: (18)'       'open: (18)'    'high: (18)'    'low: (18)'     'close: (18)'    'volum: (18)'
'14-Dec-2005'       [4.8800]        [4.9300]        [4.8100]        [4.8800]         [67064]
'15-Dec-2005'       [4.9000]        [5.0100]        [4.8800]        [4.8900]         [106106]
'16-Dec-2005'       [4.8900]        [4.9400]        [4.8400]        [4.9400]         [52277]
'19-Dec-2005'       [4.9300]        [4.9800]        [4.8700]        [4.8800]         [57681]
'20-Dec-2005'       [4.8700]        [5.0100]        [4.8600]        [4.9900]         [96808]
'21-Dec-2005'       [5.0000]        [5.0700]        [4.9500]        [4.9700]         [83140]
'22-Dec-2005'       [4.9600]        [5.0000]        [4.9300]        [4.9800]         [38033]
'23-Dec-2005'       [4.9900]        [5.0900]        [4.9800]        [5.0800]         [89875]
```

```
'26 - Dec - 2005'    [5.0900]    [5.1400]    [5.0600]    [5.0700]    [82578]
'27 - Dec - 2005'    [5.0900]    [5.0900]    [4.9800]    [5.0200]    [63578]
'28 - Dec - 2005'    [5.0200]    [5.0500]    [4.9400]    [4.9700]    [52353]
'29 - Dec - 2005'    [4.9700]    [5.1200]    [4.9500]    [5.1100]    [99265]
'30 - Dec - 2005'    [5.1100]    [5.1300]    [5.0000]    [5.0300]    [67547]
'04 - Jan - 2006'    [5.0300]    [5.1000]    [5.0200]    [5.0800]    [62509]
'05 - Jan - 2006'    [5.0900]    [5.1600]    [5.0500]    [5.1500]    [105077]
'06 - Jan - 2006'    [5.1500]    [5.2900]    [5.1200]    [5.2200]    [116407]
'09 - Jan - 2006'    [5.2200]    [5.3100]    [5.2000]    [5.2800]    [88119]
'10 - Jan - 2006'    [5.2800]    [5.3000]    [5.1500]    [5.2500]    [80245]
```

这样 at.txt 的数据内容都保存在变量 jinniu 中，在 Command 窗口下执行 whos 命令查看 jinniu。代码如下：

```
>> whos
Name          Size                Bytes  Class
jinniu        39x5                3380   fints object
```

Grand total is 329 elements using 3380 bytes

变量 jinniu 的格式是 fints 型数据，下面将日期型数据变成周类型。

```
>> weekjn = toweekly(jinniu)
```

weekjn 就是每周的时间、开盘、最高、最低、收盘及成交量。

weekjn =

```
desc:  TOWEEKLY: date  open   high   low   close   volum
freq:  Weekly (2)
'dates:(5)'        'open:(5)'   'high:(5)'   'low:(5)'   'close:(5)'   'volum:(5)'
'16 - Dec - 2005'  [2.9340]     [2.9760]     [2.9060]    [2.9420]      [45089]
'23 - Dec - 2005'  [4.9500]     [5.0300]     [4.9180]    [4.9800]      [73107]
'30 - Dec - 2005'  [5.0475]     [5.0975]     [4.9675]    [5.0325]      [70686]
'06 - Jan - 2006'  [5.0950]     [5.1700]     [5.0475]    [5.1200]      [87885]
'13 - Jan - 2006'  [5.2680]     [5.3020]     [5.1600]    [5.2560]      [8 1820]
```

将日期型数据变成月类型，只需执行下面的命令：

```
>> monthjn = tomonthly(jinniu)
    monthjn =
    desc:  TOMONTHLY: date   open   high   low   close   volum
    freq:  Monthly (3)
    'dates:(2)'      'open:(2)'   'high:(2)'   'low:(2)'   'close: (2)'   'volum:(2)'
```

```
'30 - Dec - 2005'    [4.2579]    [4.3157]    [4.2136]    [4.2671]    [6.2409e+004]
'31 - Jan - 2006'    [5.2400]    [5.2745]    [5.1320]    [5.2255]    [8.2167e+004]
```

2. 时间序列数据转化为其他类型数据

1) 时间序列数据保存为文本文件

调用方式

```
stat = fts2ascii(filename, tsobj, exttext)
```

输入参数

filename	新文件名称
tsobj	需要转化的 fints 型格式数据
exttext	不需要的描述项的行数，也可以不输入

输出参数

stat	转换成功标志：1 表示转换成功；0 表示转换不成功

例如，MATLAB 中的 fints 型变量 s 内容如下：

```
>> s
s =
    desc:  (none)
    freq:  Unknown (0)
    'dates:  (6)'        'series1:  (6)'
    '19 - Feb - 2006'    [          1.0100]
    '20 - Feb - 2006'    [          2.0200]
    '21 - Feb - 2006'    [          3.0300]
    '22 - Feb - 2006'    [          4.0400]
    '23 - Feb - 2006'    [          5.0500]
    '24 - Feb - 2006'    [          6.0600]
```

下面将变量 s 内容保存到 aa.txt 文件。

```
>> a = fts2ascii('aa.txt',s)
a =
      1
```

a=1 表示转化成功，MATLAB 的 work 目录下生成了 aa.txt 文件。用 type 命令浏览文件中的内容。

```
>> type aa.txt
```

```
dates       series1
19 - Feb - 2006      1.010000
20 - Feb - 2006      2.020000
21 - Feb - 2006      3.030000
22 - Feb - 2006      4.040000
23 - Feb - 2006      5.050000
24 - Feb - 2006      6.060000
```

2）将时间序列数据转换为矩阵数据

将 fints 型数据转化为矩阵形式的函数是 fts2mat。

调用方式

```
tsmat = fts2mat(tsobj, datesflag)
```

输入参数

tsobj	需要转换的原始数据
datesflag	0(默认值)表示不输出日期到矩阵中，1 表示日期也转换

输出参数

tsmat	转换后的矩阵

例如 s 是 fints 型数据，现转化为矩阵，如果不输入 datesflag，则日期就不会出现，则代码如下：

```
>> fts2mat(s)
ans =
      1.0100
      2.0200
      3.0300
      4.0400
      5.0500
      6.0600
```

如果将 datesflag 设为 1，则日期也被输出。代码如下：

```
>> fts2mat(s,1)
ans =
      1.0e + 005 *
      7.3273      0.0000
      7.3273      0.0000
      7.3273      0.0000
```

```
    7.3273      0.0000
    7.3273      0.0001
    7.3273      0.0001
```

输出的第一列为序数型日期。

3) fints 型数据求最大值、最小值、均值、标准差及排序

对 fints 型数据求最大值、最小值、均值、标准差及排序的函数分别是 max,min,mean,std,sortfts。

4) 实现时间序列的转换

convertto 函数实现时间序列的转换功能，功能同 toweeekly,tomonthly 等函数。

调用方式

```
newfts = convertto(oldfts, newfreq)
```

输入参数

oldfts	需要转换的数据
newfreq	转换的目标，可以根据要求选择，具体如下： 'D'或'd'为天 'W'或'w'为周 'M'或'm'为月 'Q'或'q'为季度 'S'或's'为半年 'A'或's'为年

输出参数

newfts	转换后的数据

对于前面的方法也可以用 convertto 函数实现。

如果转换成周数据，则代码如下：

```
>> weekjn = convertto(jinniu,'w')
 weekjn =
 desc:  TOWEEKLY: date    open    high    low     close    volum
 freq:  Weekly (2)
'dates:(5)'         'open:(5)'    'high:(5)'    'low:(5)'     'close:(5)'    'volum:(5)'
'16 - Dec - 2005'   [2.9340]      [2.9760]      [2.9060]      [2.9420]       [4.5089e + 004]
'23 - Dec - 2005'   [4.9500]      [5.0300]      [4.9180]      [4.9800]       [7.3107e + 004]
'30 - Dec - 2005'   [5.0475]      [5.0975]      [4.9675]      [5.0325]       [7.0686e + 004]
'06 - Jan - 2006'   [5.0950]      [5.1700]      [5.0475]      [5.1200]       [87885]
```

```
'13-Jan-2006'    [5.2680]    [5.3020]    [5.1600]    [5.2560]    [8.1820e+004]
```

转换成月数据，代码如下：

```
>> weekjn = convertto(jinniu,'m')
 weekjn =
    desc:  TOMONTHLY: date    open    high    low    close    volum
    freq:  Monthly (3)
'dates:(2)'        'open:(2)'   'high:(2)'   'low: (2)'   'close: (2)'   'volum: (2)'
'30-Dec-2005'      [4.2579]     [4.3157]     [4.2136]     [4.2671]       [6.2409e+004]
'31-Jan-2006'      [5.2400]     [5.2745]     [5.1320]     [5.2255]       [8.2167e+004]
```

5）时间序列数据的抽取

extfield 函数实现数据的抽取。

调用方式

```
ftse = extfield(tsobj, fieldnames)
```

输入参数

tsobj	原始数据
fieldnames	原始数据中的字段名

输出参数

ftse	需要的数据

如在数据 weekjn 中只要收盘价，并且命名为变量 weekjnclose，这时可以执行如下命令：

```
>> weekjnclose = extfield(weekjn,'close')
 weekjnclose =
    desc:  TOWEEKLY: date    open    high    low    close    volum
    freq:  Weekly (2)
    'dates:  (10)'      'close:  (10)'
    '16-Dec-2005'       [      2.9420]
    '23-Dec-2005'       [      4.9800]
    '30-Dec-2005'       [      5.0325]
    '06-Jan-2006'       [      5.1200]
    '13-Jan-2006'       [      5.1940]
    '20-Jan-2006'       [      5.0925]
    '27-Jan-2006'       [      5.3760]
    '03-Feb-2006'       [      5.4300]
    '10-Feb-2006'       [      5.8380]
```

```
   '17 - Feb - 2006'      [      5.9660]
```

这样 weekjn 数据中的收盘价就保存到 weekjnclose 中。

实际上在 MATLAB 中，weekjn 是一个结构数据，close 是 weekjn 的一个属性，可以用结构变量的方法查看其内容。代码如下：

```
>> weekjnclose = weekjn.close
 weekjnclose =
   desc:  TOWEEKLY: date       openhigh    low    close    volum
   freq:  Weekly (2)
   'dates:  (10)'     'close:  (10)'
   '16 - Dec - 2005'      [      2.9420]
   '23 - Dec - 2005'      [      4.9800]
   '30 - Dec - 2005'      [      5.0325]
   '06 - Jan - 2006'      [      5.1200]
   '13 - Jan - 2006'      [      5.1940]
   '20 - Jan - 2006'      [      5.0925]
   '27 - Jan - 2006'      [      5.3760]
   '03 - Feb - 2006'      [      5.4300]
   '10 - Feb - 2006'      [      5.8380]
   '17 - Feb - 2006'      [      5.9660]
```

6）将价格序列转化为收益率序列

price2ret 表示将价格序列转换为收益率序列，计算方法为 $r_t = \log(P_t/P_{t-1})$，下面是一个例子。

```
>> s = 100 * exp(0.1 * [0:5])
s =
     100.0000   110.5171   122.1403   134.9859   149.1825   164.8721
>> price2ret(s)
ans =
     0.1000     0.1000     0.1000     0.1000     0.1000
```

7）将收益率序列转化为价格序列

ret2price 可以将收益率序列转化为价格序列，方法为 $P_t = P_{t-1} \times \exp(r_t)$，下面是一个例子。

```
>> s = [0.1 0.1 0.1 0.1 0.1 0.1]
s =
     0.1000     0.1000     0.1000     0.1000     0.1000     0.1000
```

```
>> p = 100 * ret2price(s)
p =
     100.0000    110.5171    122.1403    134.9859    149.1825    164.8721    182.2119
```

3. 处理时间序列中的缺失数据

金融时间序列的数据有时会有一些数据的缺失，需要将缺失数据补齐后才能分析，fillts 函数可以利用插值的方法把缺失的数据补齐。

调用方式

```
newfts = fillts(oldfts, method)
```

输入参数

oldfts	原始数据
method	处理缺失值的方法，主要有如下几种： 'linear'或'le'线性插值法（MATLAB 默认值） 'cubic'或'c'三次插值 'spline'或's'样条法 'nearest'或'n'最近法 'pchip'或'p'逐段光滑的三次 Hemite 多项式法

输出参数

newfts	处理后的数据

下面是一个处理缺失数据的例子。

```
>> randn('seed',0);                  % 设定随机数种子为 0
>> a = randn(6,1);                   % 生成六行一列的随机数向量
>> b = [today;today + 5]'            % 从今天到后面五天
b =
     732979
     732980
     732981
     732982
     732983
     732984
>> fts = fints(b,a)                  % 生成 fints 格式数据
fts =
        desc:   (none)
        freq:   Unknown (0)
```

```
        'dates:(6)'        'series1:(6)'
        '29-Oct-2006'      [1.1650]
        '30-Oct-2006'      [0.6268]
        '31-Oct-2006'      [0.0751]
        '01-Nov-2006'      [0.3516]
        '02-Nov-2006'      [-0.6965]
        '03-Nov-2006'      [1.6961]
>> fts(3) = NaN;                    % 将第三个数据变为缺失值 NaN
>> fts(3)                           % 查看改变后的 fts 变量中第三个数据的值
ans =
      desc:  (none)
      freq:  Unknown (0)
      'dates:(1)'      'series1:(1)'
      '31-Oct-2006'    [NaN]
>> newdata = fillts(fts,'linear')% 调用 fillts 函数中的线性插值处理 fts 中的缺失值
newdata =
          desc:  Filled
          freq:  Unknown (0)
          'dates:(6)'        'series1:(6)'
          '29-Oct-2006'      [1.1650]
          '30-Oct-2006'      [0.6268]
          '31-Oct-2006'      [0.4892]
          '01-Nov-2006'      [0.3516]
          '02-Nov-2006'      [-0.6965]
          '03-Nov-2006'      [1.6961]
```

从上面的结果可以知道,缺失值变为 0.489 2。

思考题

1. 将金牛股份(000937)2005 年 12 月 15 日至 2006 年 1 月 6 日的日收益率数据保存在 jinniu.txt 文件中。计算期间的最大跌幅。

2. 计算金牛股份(000937)2005 年 12 月 15 日至 2006 年 1 月 6 日的日收益率均值、标准差及夏普比率。把结果分别用 TXT 文件和 Excel 文件保存。

第 7 章　MATLAB 和其他软件及网站的数据连接

MATLAB 开发了和其他软件的接口，使得 MATLAB 像一个平台，通过这个平台来实现数据间的交换。本章要求读者了解 MATLAB 和 Excel 之间数据交换的途径，学会从 yahoo 等网站获取国内外债券市场与股票市场历史数据的方法；了解 Word 环境下启动 MATLAB 方法。

7.1　MATLAB 和 Excel 的数据连接

MATLAB 提供了文件读写函数，这些函数是 MATLAB 语言的一部分，也不需要专门工具箱支持。MATLAB 文件读写函数分为高级函数和低级函数。高级函数调用方法简单，使用方便，但是灵活性差，仅仅适用于一般格式文件类型。低级函数比较烦琐，但是灵活性好。

MATLAB 应用程序接口 API（Application Program Interface）所涵盖的内容非常广泛，可以和多种语言实现数据连接，如 C＋＋，Excel，Word，Spss，VB，而且可以与 Java 等网络语言实现数据连接。鉴于市面上已经有许多书籍介绍 MATLAB 与 C 语言、Word、VB、Java 软件的数据连接，这里就不再赘述。下面主要介绍 MATLAB 与 Excel，Word 的数据连接。

Excel 和 MATLAB 在数据显示和数值计算上各有优势。Excel 是商业运用最广泛的工具，非常直观，但是数值编程比较差，而 MATLAB 可以弥补这一点，有时在程序开发上需要将两者结合起来，实现二者之间的优势互补。为此 MATLAB 提供了 Excel Link 和 MATLAB Builder For Excel 两个连接工具，实现 MATLAB，Excel，VBA 之间混合编程。

Excel Link 是一个在 Windows 环境下实现 MATLAB 和 Excel 互相连接的插件。通过对 Excel 和 MATLAB 连接，用户可以在 MATLAB 工作区中使用 Excel 宏编写程序，使用 MATLAB 数据处理和图形图像处理功能进行相关操作，同时保证两个工作环境数据连接和同步。使用 Excel Link 时不必离开 Excel 环境，直接在 Excel 工作区调用 MATLAB 变量即可。

对于 Excel 而言，Excel Link 也可以视为一个加载宏（ExcelAddin）。加载宏是一种特殊的程序，通常用来增加 Excel 的功能。加载宏对于用户而言不可见，可以通过 VBA 的代码访问。加载宏的来源有 3 种：一是 Excel 自带的加载宏；二是 Excel 中录制宏插件形成的加载宏；三是其他语言编制的加载宏。

7.1.1　加载 Excel Link

Excel 加载 Excel Link 的步骤如下：

① 打开 Excel 软件，打开【工具】菜单，选择【加载宏】菜单项，弹出【加载宏】对话框。

② 在【加载宏】对话框中单击【浏览】按钮，在 MATLAB/toolbox/exlink 文件夹下选中 Excllink 文件，单击【确定】按钮。

③ 重新回到 Excel 中【加载宏】对话框，在“Excel Link 2.2 for use with MATLAB”选项前打钩，如图 7.1 所示。



图 7.1　Excel 工具菜单下加载宏

此时发现 MATLAB 命令窗口已经加在 Excel 任务窗口中，单击【startmatlab】按钮就会启动【MATLAB Command Window】窗口，如图 7.2 所示。

图 7.3 是 Excel 中操作 MATLAB 命令菜单。

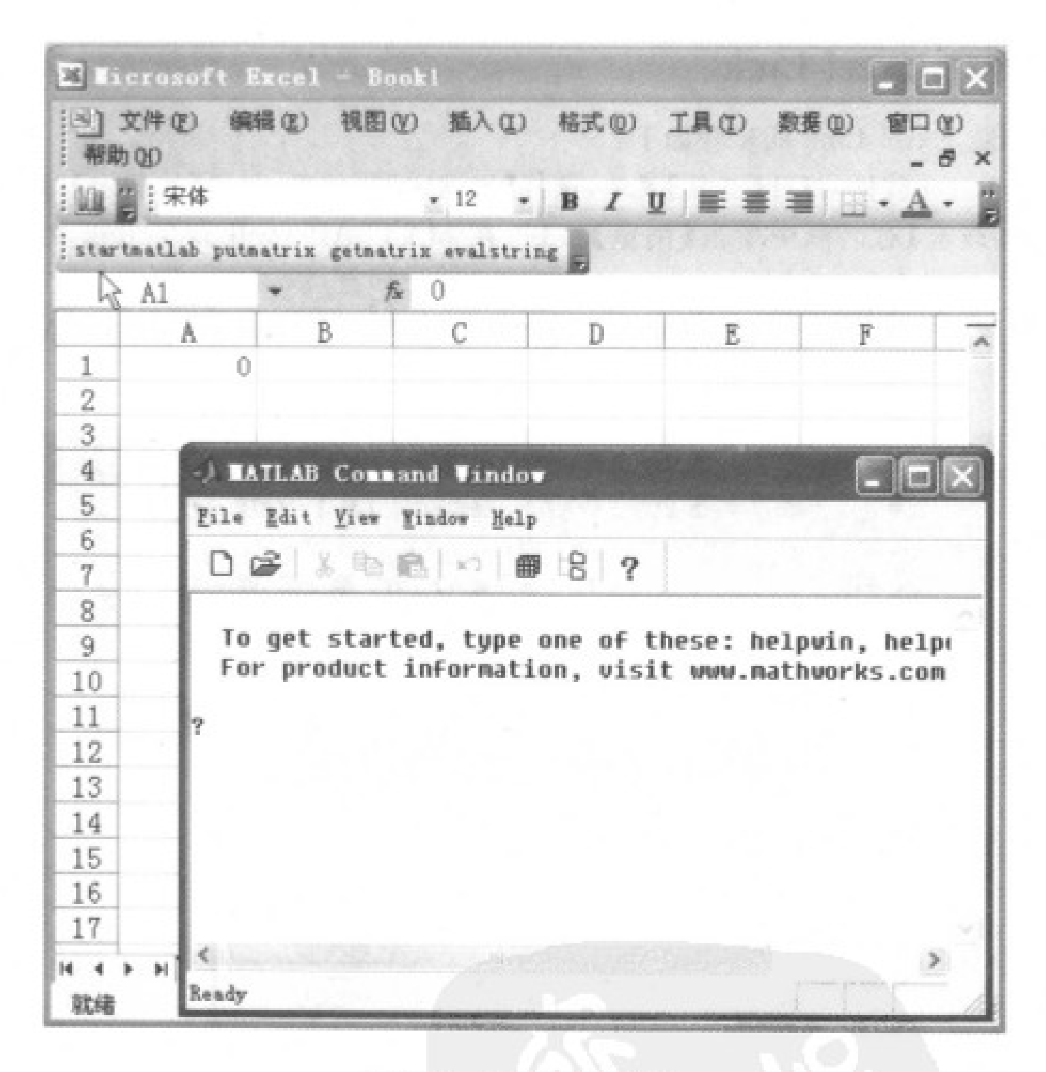


图 7.2 加载宏后的界面

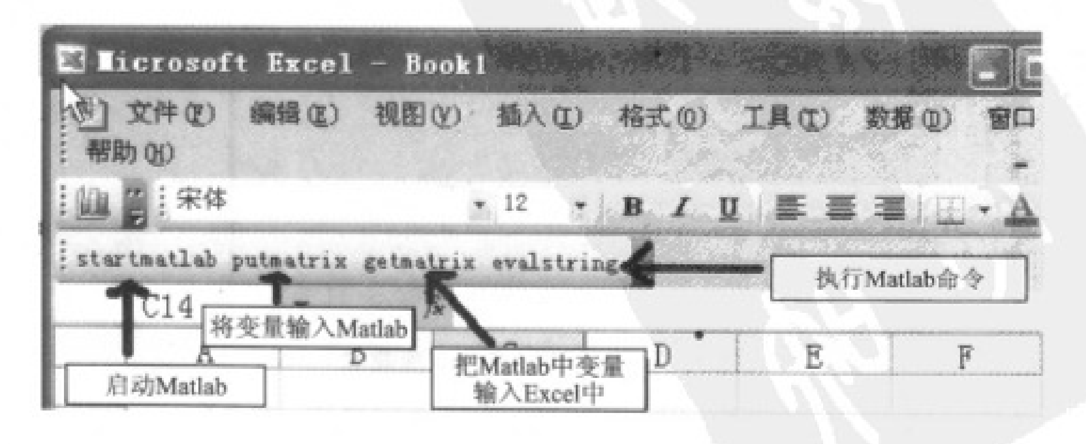


图 7.3 Excel 中 MATLAB 命令菜单

窗口中的 4 个按钮【startmatlab】、【putmatrix】、【getmatrix】、【evalstring】用以实现 Excel

与 MATLAB 实时数据交换。

7.1.2　MATLAB 自动启动和 Excel 连接

在桌面 MATLAB 7 小图标下右击，弹出【属性】快捷菜单，在对话框中【目标】文本框的“C:\MATLAB7\bin\win32\MATLAB.exe”后面加入“ /automation”，具体如图 7.4 所示。



图 7.4　MATLAB 和 Excel 自动连接

这样单击 MATLAB 快捷键就自动出现与 Excel 连接的窗口。如果要关闭【MATLAB Command Window】窗口，同样进入 Excel 中的【工具】菜单，在【加载宏】窗口去掉“Excel Link 2.2 for use with MATLAB”选项前小钩，就取消 Excel 与 MATLAB 连接。

7.1.3 Excel Link 的使用

1. 连接管理函数

通过 Excel Link 函数，Excel 就成了 MATLAB 一个功能强大的数据存储和应用终端，使用户不必脱离 Excel 环境，而只要以数据表单元的函数形式或宏命令来使用 MATLAB 提供的相关处理函数即可。

Excel Link 提供了 4 个连接函数，分别是 MATLAB 初始化、启动、打开和终止，其中，MATLABinit 只能以宏命令方式运行，其他命令均可以作为数据单元函数或者宏命令执行。

MLAutoStart 用来设置自启动，系统默认设置是在每一次启动 Excel 时自动启动 MATLAB，如果选择手动启动，就需要 MATLABinit 初始化 Excel Link 并且启动 MATLAB；MLClose用来在保持 Excel 继续运行状态下终止 MATLAB 运行；MLOpen 或者 MATLABinit 在原来 Excel 进程中重新启动 MATLAB。具体内容如表 7.1 所列。

表 7.1 MATLAB 中启动 Excel 命令

函 数	作 用	函 数	作 用
MATLABinit	初始化 Excel Link，启动 MATLAB	MLClose	终止 MATLAB
MLAutoStart	自动启动 MATLAB	MLOpen	启动 MATLAB

除了上述 4 个连接函数，MATLAB 还提供了 9 个数据管理函数，实现 Excel 和 MATLAB 之间数据连接，并可在 Excel 中执行 MATLAB 命令，具体内容如表 7.2 所列。

表 7.2 Excel 中执行 MATLAB 命令

函 数	作 用
MATLABfcn	对于给定的 Excel 数据运行 MATLAB 命令
MATLABsub	对于给定的 Excel 数据运行 MATLAB 命令，并指定输出位置
MLEDeleteMatrix	删除 MATLAB 矩阵
MLEvalString	执行 MATLAB 命令
MLGetMatrix	向 Excel 数据表写入 MATLAB 数据
MLAppendMatrix	向 MATLAB 工作区添加 Excel 数据
MLPutMatrix	用 Excel 数据表创建或覆盖 MATLAB 数据
MLPutVar	用 Excel 数据表 VBA 创建或覆盖 MATLAB 数据
MLGetVar	向 Excel 数据表 VBA 写 MATLAB 数据内容

用户能以数据表单元函数形式或者宏命令形式调用除 MLGetVar 和 MLPutVar 以外所有数据处理函数，MLGetVar 和 MLPutVar 只能以宏命令形式被调用。

如果出现错误，则 MATLAB 会自动在 Excel 单元格中给出错误信息，便于使用者进行核查。

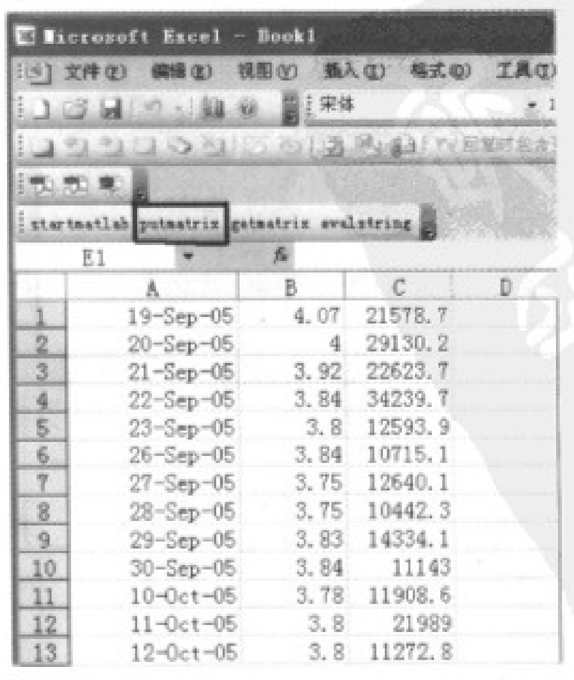
MATLAB 连接错误汇总如下：

#COLS>256　　变量名超出 256
#COMMAND!　　无法识别的命令
#DIMENSION!　　使用添加数据操作函数，但所添加数据的维数和原数据不符
#INVALIDNAME!　　无效的变量名
#INVALIDTYPE!　　无效的数据类型
#MATLAB?　　MATLAB 未运行
#NAME?　　无法识别函数名
#NONEXIST!　　引用了不存在的矩阵名称
#ROWS>65536　　变量行数超出限制
#SYNTAX?　　语法错误
#VALUE!　　参数缺少或类型错误

2. 直接把数据输入到 MATLAB 中

例如把 Excel 中厦门建发(600153)从 2005 年 9 月 19 日到 2006 年 6 月 19 日数据录入 MATLAB 中，保存变量名为 xmjf_600153，具体步骤如下。

第一步：Excel 中厦门建发数据分别为日期、收盘价和成交量，具体如图 7.5 所示。



	A	B	C	D
1	19-Sep-05	4.07	21578.7	
2	20-Sep-05	4	29130.2	
3	21-Sep-05	3.92	22623.7	
4	22-Sep-05	3.84	34239.7	
5	23-Sep-05	3.8	12593.9	
6	26-Sep-05	3.84	10715.1	
7	27-Sep-05	3.75	12640.1	
8	28-Sep-05	3.75	10442.3	
9	29-Sep-05	3.83	14334.1	
10	30-Sep-05	3.84	11143	
11	10-Oct-05	3.78	11908.6	
12	11-Oct-05	3.8	21989	
13	12-Oct-05	3.8	11272.8	

图 7.5　Excel 中厦门建发股价数据

第二步：单击【startmatlab】按钮启动 MATLAB，出现界面如图 7.6 所示。



图 7.6　启动 MATLAB 界面

第三步：单击【putmatrix】按钮后将会出现一个菜单，其界面如图 7.7 所示。



图 7.7　命名股价变量

根据菜单提示将 Excel 中数据区用光标选定，传输到 MATLAB 中变量名为 xmjf_600153，然后单击【确定】按钮，则 MATLAB 中就会出现 xmjf_600153 变量，Excel 中数表被储存在变量 xmjf_600153 中，MATLAB 工作区内容如图 7.8 所示。



图 7.8　厦门建发在 MATLAB 中的变量

变量 xmjf_600153 第一列是整数，38 614 表示 2006 年 6 月 19 日距 1900 年 1 月 1 日为 38 614天，MATLAB 中时间起始为 0000 年 1 月 1 日，和 Excel 中时间相差 693 960 天。

3. MATLAB 中自带 Excel 文件介绍

【例 7-1】　打开 MATLAB 的 toolbox 中 exlink 目录下的 ExliSamp.xls 文件，如图 7.9 所示。

下面分别介绍文件内容，打开 sheet1，依次运行 7 个步骤。

第一步：将 Excel 中 DATA 数据传给 MATLAB 工作区中变量 data，变量 data 储存了 3 个变量 25 个观察值。

Regression and Curve Fitting

DATA

35	207	1325
17	90	533
43	180	1013
458	180.5621	402.008
476	220.781	515.8528
495	354.7157	549.7114
521	362.1239	543.0184
532	421.8399	524.5499
533	449.7776	513.775
543	379.0353	522.2081
602	508.0868	554.761
635	506.0071	611.0947
671	692.2141	686.9715
766	801.4852	775.6072
913	892.428	869.023
938	945.09	959.3974
1013	1217.144	1040.419
1038	1344.976	1108.636
1134	1181.723	1164.812
1163	1247.572	1215.276
1319	1616.727	1273.275
1325	1448.474	1360.322
1591	2112.748	1507.557
2006	2151.934	1757.09
2043	2672.26	2163.358
2904	2690.806	2794.475
3282	3462.625	3733.586
5326	4365.865	5080.215

Excel Link Functions

1. Transfer the data to MATLAB.
   0 <== MLPutMatrix("data",DATA)

2. Set up data for regression.
   0 <== MLEvalString("y = data(:,3)")
   0 <== MLEvalString("e = ones(length(data),1)")
   0 <== MLEvalString("A = [e data(:,1:2)]")

3. Compute regression coefficients.
   0 <== MLEvalString("beta = A\y")

4. Calculate regressed result.
   0 <== MLEvalString("fit = A*beta")

5. Compare original data with regression results.
   0 <== MLEvalString("[y,k] = sort(y)")
   0 <== MLEvalString("fit = fit(k)")
   0 <== MLEvalString("n = size(data,1)")

6. Use MATLAB's polynomial solving functions for anoth
   0 <== MLEvalString("[p,S] = polyfit(1:n,y',5)")
   0 <== MLEvalString("newfit = polyval(p,1:n,S)")

7. Plot curves and add legend
   0 <== MLEvalString("plot(1:n,y,'bo',1:n,fit,'r:',1:

图 7.9 Excel 数据拟合

第二步：在 Excel 中执行 MATLAB 命令。

```
MLEvalString("y = data(:,3)")              % 将变量 data 中第三列作为因变量
MLEvalString("e = ones(length(data),1)")
MLEvalString("A = [e data(:,1:2)]")
```

第三步：计算回归系数。

第四步：计算回归结果。

第五步：将原结果和回归结果相比较。

第六步：用 MATLAB 中多项式进行拟合。

第七步：对拟合结果作图。

```
MLEvalString("plot(1:n,y,'bo',1:n,fit,'r:',1:n,newfit,'g'); legend('data','fit','newfit')")
```

运行结果如图 7.10 所示。

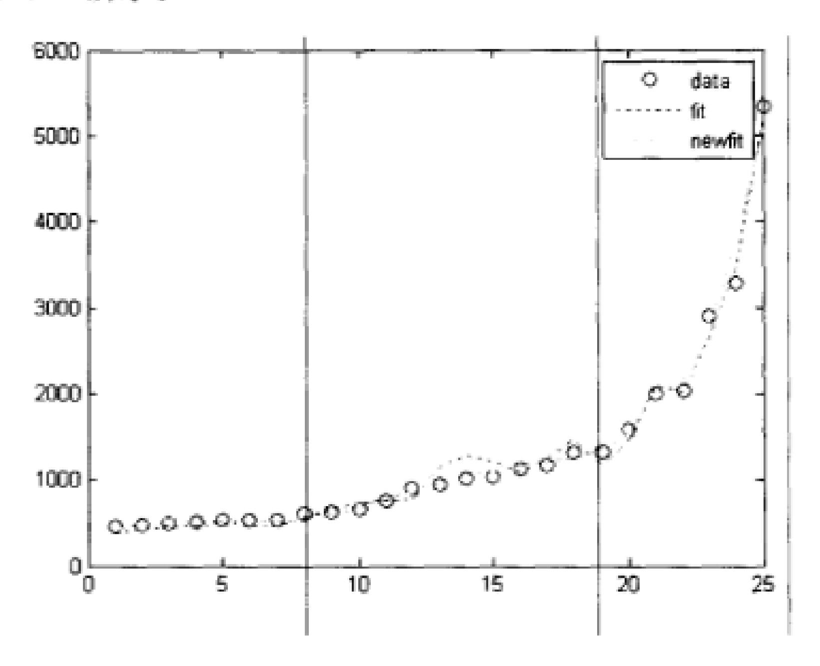


图 7.10　数据拟合图

下面在 Excel 中继续调用 MATLAB 程序，用二叉树方法计算欧式看跌期权价格。

第一步：打开“Sheet4”数据 B4:B10 名称是 bindata，分别存放股票价格、行权价、无风险利率、期权存续期(5/12)、时间离散步长(1/12)、股票标准差以及是否是欧式(美式)期权，B15 开始存放二叉树各节点价格 asset_tree，B23 开始存放现金流 value_tree。

第二步：激活 D5 单元，将 Excel 中 bindata 传到 MATLAB 中变量“b”中。激活 D12 单元，调用 MATLAB 中计算二叉树函数“binprice”，其二叉树数据结果保存在变量“p”中，价格保存在变量“o”中。激活 D11，D12，分别保存在 MATLAB 中变量 asset_tree 与 value_tree 中，注意 B15 与 B23 中分别是“asset_tree”与“value_tree”的起点。

下面计算资产组合有效前沿。

第一步：打开“Sheet5”，激活 A15，将 MATLAB 中 F3:G3 数据传到 MATLAB 中变量 labels。激活 B4:D9 数据，并将之传到 MATLAB 中，命名为变量“retseries”。

第二步：激活 A15，根据输入时间序列计算收益率与方差。激活 A20，执行 MATLAB 中的命令。

第三步：将变量输出到 Excel 中，风险(risk)输出到 F4，期望收益输出到 G4，资产权重输出到 H4 中。

第四步：调用 MATLAB 函数，绘制资产组合有效前沿，如图 7.11 所示。

下面计算债券现金流和现金流之间时间映射。

第一步：打开“Sheet6”，分别将 Excel 中数据传到 MATLAB 中。

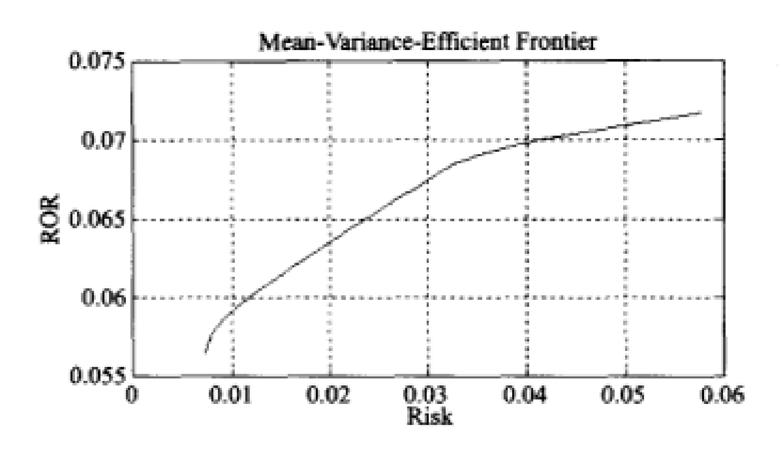


图 7.11 资产组合有效前沿

激活 A18，将 Excel 中到期日变量 Maturity 传给 MATLAB 中变量"maturity"；激活 A19，将 Excel 中债券利率传到 MATLAB 中变量"cpnrate"；激活 A20，将 Excel 中债券结算日传给 MATLAB 中变量"sd"。

第二步：调用 MATLAB 中现金和时间映射函数。

执行 A23，将 Excel 中日期型数据转换为对应 MATLAB 中的日期；执行 A24，将债券转换为现金流。

第三步：调用 MATLAB 将日期型数据转换为单元数据。

第四步：将现金流数据传到 Excel 中。

第五步：画出现金流图，如图 7.12 所示。

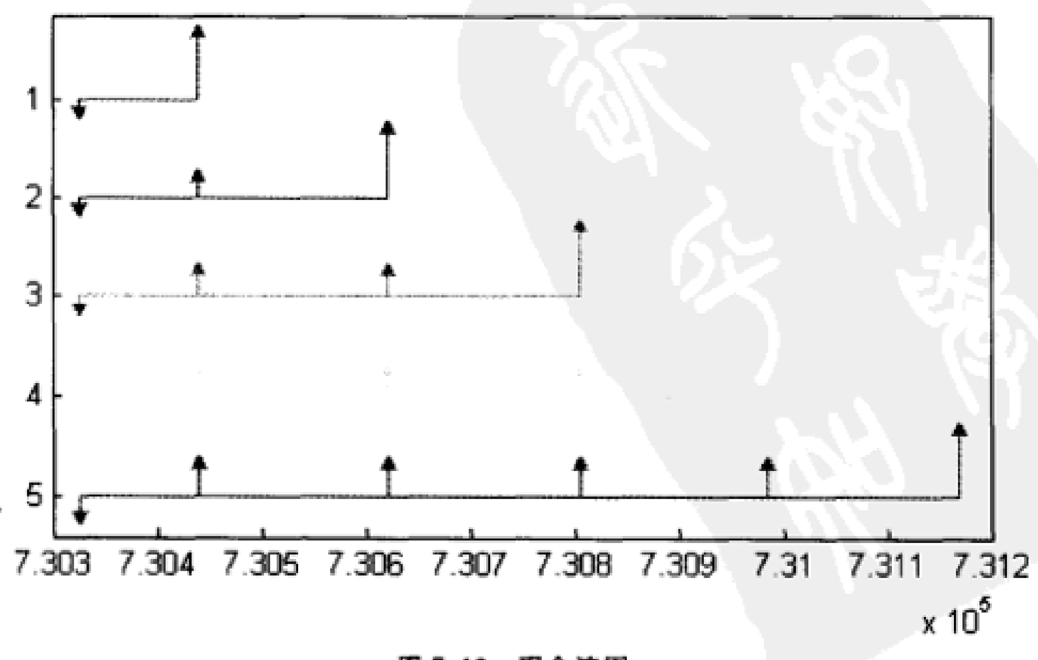


图 7.12 现金流图

4. 简易方法将 Excel 数据导入 MATLAB

首先在 Excel 中建立一个数据文件如图 7.13 所示。

Microsoft Excel - magic

文件(F)　编辑(E)　视图(V)　插入(I)　格式(O)　工具(T)　数

D8

	A	B	C	D	E
1	a	b	c	d	
2	16	2	3	13	
3	5	11	10	8	
4	9	7	6	12	
5	4	14	15	1	
6					

图 7.13　在 Excel 中建立数据文件

然后在 MATLAB 文件目录浏览器下找到 magic 文件，如图 7.14 所示。

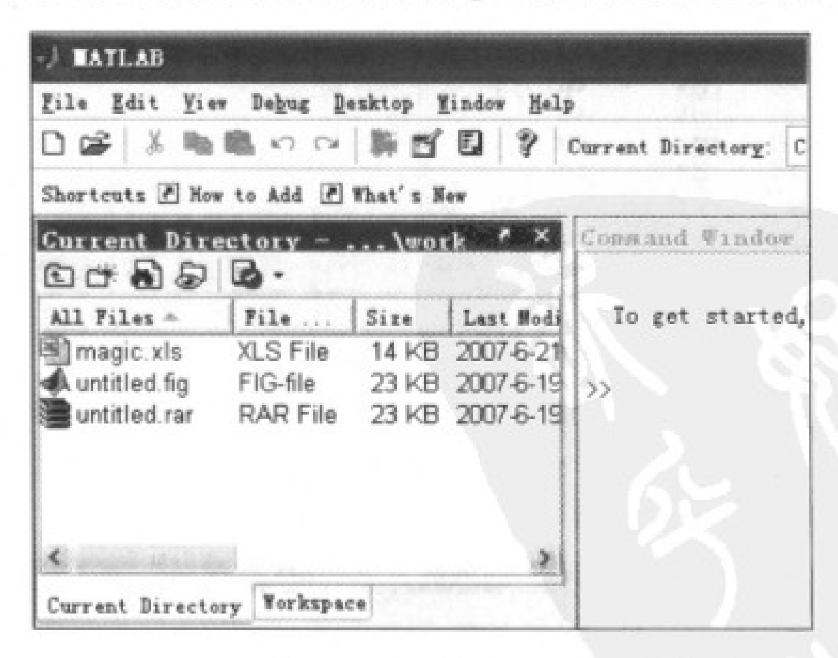


图 7.14　在 MATLAB 中调入 magic. xls 文件

用鼠标移动到 magic 文件上，右击，弹出的菜单如图 7.15 所示。

单击数据输入选项后，出现界面如图 7.16 所示。

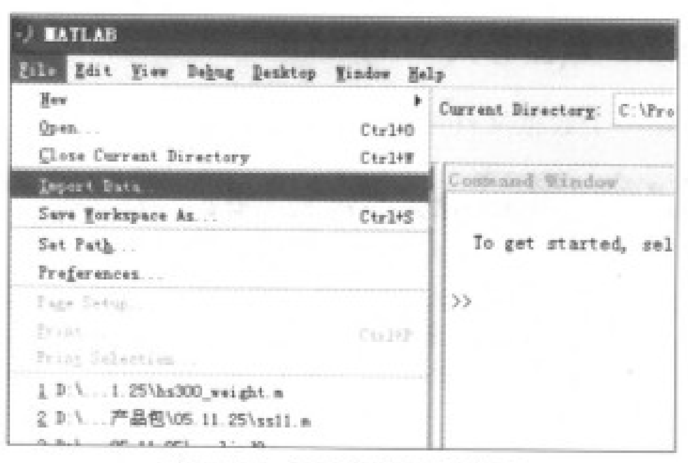


图 7.15　单击数据输入选项

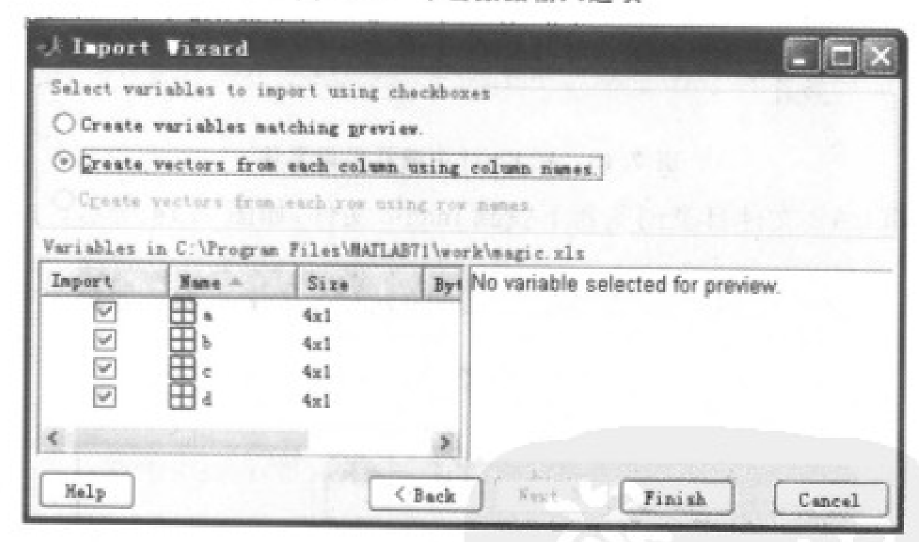


图 7.16　选择需要变量

上面的按钮表示按类型保存，数据与字符串分开保存。下面的按钮表示每列作为一个变量，第一字母为变量名，现在选择 a，b，c，d 作为变量名，只需在其前面个打个钩即可，在工作区中可以浏览到这 4 个变量，如图 7.17 所示。

Workspace　Stack: Base

Name	Value	Class
a	[16;5;9;4]	double
b	[2;11;7;14]	double
c	[3;10;6;15]	double
d	[13;8;12;1]	double

图 7.17　浏览选择变量

5. 将 Excel 日期转化为 MATLAB 日期

在 MATLAB 中将 Excel 日期转化为 MATLAB 日期的函数是 x2mdate。

调用方式

```
MATLABDate = x2mdate(ExcelDateNumber, Convention)
```

输入参数

ExcelDateNumber	MATLAB 日期
Convention	转化方式。如果等于 0,表示 Excel 起始日期为 1900 年 1 月 1 日,默认值是 0;如果等于 1,则表示起始日为 1904 年 1 月 1 日

```
>> qw                          % Excel 中 2007 年 1 月 1 日序数型日期
qw =
      39083
>> x2mdate(qw)
ans =
      733043
>> datestr(ans)
ans =
      01 - Jan - 2007
```

7.2 MATLAB 与财经网站的数据连接

MATLAB 中带有 Datafeed 工具箱(Datafeed Toolbox),该工具箱可以实现 MATLAB 与财经网站数据连接,从财经网站上下载财经数据。财经网站包括 bloomberg,factset,byperfeed,idc,yahoo 等著名专业网站。

如果和 bloomberg,factset, byperfeed,idc 财经数据公司网站实现数据交流,必须要安装客户端 API 插件,如果安装不正确,则得不到财经数据。和 FactSet FAST 建立数据连接还需要得到该网站许可,具体可查看公司网站 http://www.factset.com。

7.2.1 获得 bloomberg 网站数据

bloomberg 是世界上著名财经数据公司,可以提供各种类型财经数据,如果和 bloomberg 网站建立连接,需要调用 MATLAB 中的 bloomberg 函数。

调用方式 1

```
Connect = bloomberg
```

输入参数

bloomberg	MATLAB默认值端口是8194,网址是MATLAB中bloomberg网站默认网址

调用方式2

```
connect = bloomberg(PortNumber, IPAddress)
```

输入参数

PortNumber	机器端口
IPAddress	字符串,财经网站Internet网址

输出参数

connect	用端口建立和bloomberg数据服务器的连接

下面举例说明如何和bloomberg网站实现数据连接。首先从bloomberg公司网站上下载API软件,并完成安装。

```
>> c = bloomberg
  c =
      connection: 20135448
      ipaddress: []
      port: 8194
```

如果查看端口号、版本等信息,则可以调用get函数。

```
>> p = get(c,{'Port';'Version'})          % 查看连接c中端口号、版本
p =
    port: 8194
    version: 1.9000
```

端口号为8194,版本为1.9。

进一步地接收bloomberg数据就要使用连接函数fetch。

调用方式

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'HEADER', 'Flag','Ident')
data = fetch(Connect, 'Security', 'GETDATA', 'Fields', 'Override', 'Ident', 'Values')
data = fetch(Connect, 'Security', 'TIMESERIES', 'Date', 'Minutes', 'TickField')
data = fetch(Connect, 'Security', 'HISTORY, 'Fields', 'FromDate', 'ToDate',
         'Period', 'Currency', 'Ident')
ticker = fetch(Connect, 'SearchString', 'LOOKUP', 'Market')
```

```
data = fetch(Connect, 'Security', 'REALTIME', 'Fields', MATLABProg)
data = fetch(Connect, Security, 'STOP')
```

输入参数

Connect	bloomberg 函数输出值
Security	bloomberg 网站服务器中股票名称，是一个字符串，也可以是一个或多个股票名称构成的单元变量
Flag	默认值是最近买入价、卖出价及成交价。如 Flag = today，表示今天的价格
Currency	(Optional)货币的回报，相关货币名称在@bloomberg/bbfields.mat 中，默认值为[]
Ident	(Optional)证券类型标识符，证券类型保存在数据文件 bbfields.mat 中，默认值为[]
Fields	字段名，具体内容见 bbfields.mat，默认值为[]
Override	(Optional)Override 字段名清单，字符串或者单元变量，默认值为[]
Values	(Optional)保存 Override 的值
Date	证券时间序列日期
Minutes	(Optional)间隔分钟
TickField	(Optional)字符串或者数值，例如 TickField = 'Trade'或者 TickField = 1 将返回交易数据，用函数 dftool('ticktypes')将返回交易日字段
FromDate	历史数据开始时间
ToDate	历史数据结束时间
Period	(Optional)数据类型，'d' 日数据(默认值)，'w' 周数据，'m' 月数据，'q'季度数据，'y'年数据
Currency	(Optional) 货币名称，在文件@bloomberg/bbfields.mat 中列出了名称
Market	金融市场名称，Comdty 为商品市场，Corp 表示公司债，Curncy 表示货币，Equity 表示证券市场，Govt 表示政府债，Index 表示指数，Mkt 表示货币市场证券，Mtge 表示抵押证券，Muni 表示市政债券市场，Pfd 表示优先股市场

如果需要获得证券文件头，则在 Command 窗口中执行：

```
>> D = fetch(C,'ABC US Equity')
```

如果需要获得 ABC 股票开盘价、收盘价，则在 Command 窗口中执行：

```
>> D = fetch(C,'ABC US Equity','GETDATA',{'Last_Price';'Open'})
```

如果需要获得 ABC 股票当日时间序列，则在 Command 窗口中执行：

```
>> D = fetch(C,'ABC US Equity','TIMESERIES', now)
```

如果需要获取证券历史月收盘价，时间从 1999 年 8 月 1 日到 2000 年 9 月 30 日，则执行：

```
>> D = fetch(C,'ABC US Equity','HISTORY','Last_Price','8/01/99', '9/30/00','m')
```

如需获得 IBM 股票 1999 年 11 月 16 日时间序列，则可执行下面命令：

```
>> data = fetch(c1, 'IBM US Equity', 'TIMESERIES', '11/16/99')
```

结果如下：

```
31.00    730440.31    130.00    1000.00
32.00    730440.31    130.00     200.00
32.00    730440.35    129.50   10000.00
31.00    730440.35    129.50     100.00
32.00    730440.35    129.50     100.00
 1.00    730440.56    129.25    4000.00
31.00    730440.56    129.38    1500.00
32.00    730440.56    129.50     500.00
 1.00    730440.56    129.63    5000.00
31.00    730440.56    129.63     400.00
32.00    730440.56    129.63     200.00
 1.00    730440.56    129.69    5000.00
31.00    730440.56    129.69     500.00
32.00    730440.56    129.69     500.00
31.00    730440.56    129.75     100.00
32.00    730440.56    130.00     100.00
 1.00    730440.56    130.00    5000.00
 1.00    730440.56    129.88    5000.00
31.00    730440.56    129.88     300.00
```

第一列是 Flag 值，第二列是时间序列，第三列是成交价格，第四列是成交量。

如果需要获取 IBM 股票历史上收盘价，时间是 1999 年 7 月 15 日到 8 月 2 日，则执行如下命令：

```
>> data = fetch(c1, 'IBM US Equity', 'HISTORY', 'Last_Price', '07/15/99', '08/02/99')
data =
730316.00    136.31
730317.00    136.25
730320.00    134.63
730321.00    128.25
730322.00    129.00
730323.00    123.88
730324.00    124.81
730327.00    123.00
730328.00    126.25
```

```
730329.00          128.38
730330.00          125.38
730331.00          125.69
/730334.00         122.25
```

第一列是日期，第二列是收盘价。

确定是否已经和 bloomberg 网站连接可以执行下面命令：

```
>> x = isconnection(Connect)
```

如果返回值为 1，则表示已经连接上；0 表示没有连接。

如果想关掉和 MATLAB 连接，则可以执行下面命令：

```
>> close(Connect)
```

7.2.2 获得 yahoo 网站数据

1. 建立和 yahoo 网站服务器连接

调用方式

```
Connect = yahoo
Connect = yahoo('URL', 'IPAddress', PortNumber)
```

输入参数

URL	yahoo 网站网址
IPAddress	代理服务器 IP 地址
PortNumber	代理服务器端口

建立和 yahoo 网站的联系可以执行如下命令：

```
>> Connect = yahoo('http://quote.yahoo.com','111.222.33.444',5678)
```

2. 获得 yahoo 网站数据

调用方式

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Date')
data = fetch(Connect, 'Security', 'Fields', 'Date')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate')
```

```
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')
```

输入参数

Connect	yahoo函数输出值
Security	证券名称，如IBM股票可以执行'IBM'，yahoo网站每次只能输入一只股票
Data	日期
FromData	证券历史数据开始日期
ToData	证券历史数据终止日期
Period	选取数据类型，'d'表示日数据，'w'表示周数据，'m'表示月数据
Fields	字符串，例如Symbol表示证券名称，Last表示上一个交易日数据，Data表示日期，Time表示时间，Open表示开盘价，High表示最高价，Low表示最低价，Volume表示成交量

例如和yahoo网站进行连接可以执行命令：

```
>> Connect = yahoo
Connect =
      url: 'http://quote.yahoo.com'
       ip: []
     port: []
```

如果要进一步查询可口可乐2000年4月6日收盘价，则可以执行如下命令：

```
>> ClosePrice = fetch(Connect,'ko','Close','Apr 6 00')
ClosePrice =
              730582     45.75
```

从上面可以看出可口可乐收盘价为45.75美元。

如果今天是2006年11月6日，需要查看IBM股票上最新收盘价，则执行如下命令：

```
>> ClosePrice = fetch(Connect,'IBM','last')
ClosePrice =
            Last: 91.4100
```

可以知道IBM股票最新收盘价为91.41美元。

如果要查询上海市场的股票价格只需在代码后面加上SS，深证加上SZ。例如今天是2008年8月30日16点，如果查询当天深发展A(000001.SZ)的价格，则可以执行如下命令：

```
>> Price = fetch(Connect,'000001.SZ')
Price =
    Symbol: {'000001.SZ'}
```

```
      Last: 37.7000
      Date: 733284
      Time: 7.3304e+005
    Change: 0.8600
      Open: 37.2800
      High: 38.0800
       Low: 36.5800
    Volume: 24682722
>> datedisp(Price.Date)
30-Aug-2007
```

可以知道深发展 A(000001.SZ)2007 年 8 月 30 日最新价(last)是 37.70 元,涨了 0.86 元(change),开盘(open)37.28 元,最高(high)38.08 元,最低价(last)36.58 元,成交量(volume)24 682 722 股。

```
>> datestr(Price.Time,'mmm.dd.yyyy HH:MM:SS')
ans =
      Jan.01.2007 03:00:00
```

需要注意的是由于中国和美国存在 12 个小时的时差,美国时间 2007 年 8 月 30 日上午 3 点是中国时间下午 3 点。

附录 6 列举了 yahoo 网站支持的交易所,附录 7 列举了美国股票市场的股票代码。

如果查看 MATLAB 是否与网站建了连接,则可以输入如下命令:

```
>> isconnection(Connect)
ans =
      1
```

返回值是 1,说明已经建立连接;如果返回值是 0,则没有建立连接。如果需要取消一个连接,则可以执行如下命令:

```
>> close(Connect)
```

在 MATLAB 中用 get 函数查看连接性质,如果查看连接网址,则可以执行如下命令:

```
>> get(Connect,'url')
ans =
      http://quote.yahoo.com
```

则显示连接网址是 http://quote.yahoo.com。

7.2.3 获得 FactSet 网站数据

调用方式

```
Connect = factset('UserName','SerialNumber','Password', CustomerID')
```

输入参数

UserName	用户注册名
SerialNumber	用户序列号
Password	密码
CustomerID	FactSet 用户 ID

如果要获取 Connect 内容，则可以用 get 函数。

调用方式

```
value = get(Connect, 'PropertyName')
```

输入参数

PropertyName	Connect 的相关属性，属性包括 user,serial,password,cid

下面建立和 FactSet 的连接。

```
>> Connect = factset('Fast_User','1234','Fast_Pass','userid')
>> h = get(Connect)
h =
      user: 'Fast_User'
      serial: '1234'
      password: 'Fast_Pass'
      cid: 'userid'
```

判断 MATLAB 有没有和 FaceSet FAST 建立联系，则可以执行下列命令：

```
>> x = isconnection(Connect)
```

1 表示已经连接，0 表示没有连接。

如果关闭和 FaceSet FAST 之间联系，则可以执行下列命令：

```
>> close(Connect)
```

7.2.4 获得 Hyperfeed 中的数据

调用方式

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Date')
data = fetch(Connect, 'Security', 'Fields', 'Date')
```

```
data = fetch(Connect, 'Security', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')
```

输入参数

Connect	MATLAB 中 hyperfeed 函数输出值
Security	证券名称
Fields	字符串或单元变量。其取值为 Symbol,LastDate,Time,Change,OpenHigh,Low,Volume
Date	日期字符串,或含有日期字符串
FromDate	开始日期
ToDate	最后日期
Period	股票数据类型。可以选择的有,d(日数据)、w(周数据)、m(月数据)、v(单个数据)

如要获得 Coca Cola 公司 2000 年 4 月 6 日收盘价,则可以执行如下命令:

```
>> c = hyperfeed('History');
>> ClosePrice = fetch(c,'ko','Close','Apr 6 00')
ClosePrice =
            730582.00          45.75
```

可知 2000 年 4 月 6 日可口可乐收盘价是 45.75 美元。

7.2.5 建立和 FT 服务器的连接

调用方式

```
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate', 'Period')
data = fetch(Connect,'','GUILookup','GUICategory')
```

输入参数、输出参数同前。

7.2.6 MATLAB 和财经网站数据接口 GUI

MATLAB 软件为了和财经类数据网站更好地进行数据交流,开发了 GUI 进行数据交换。

打开 GUI,在 Command 窗口中执行如下命令:

```
>> dftool
```

GUI 界面如图 7.18 所示。

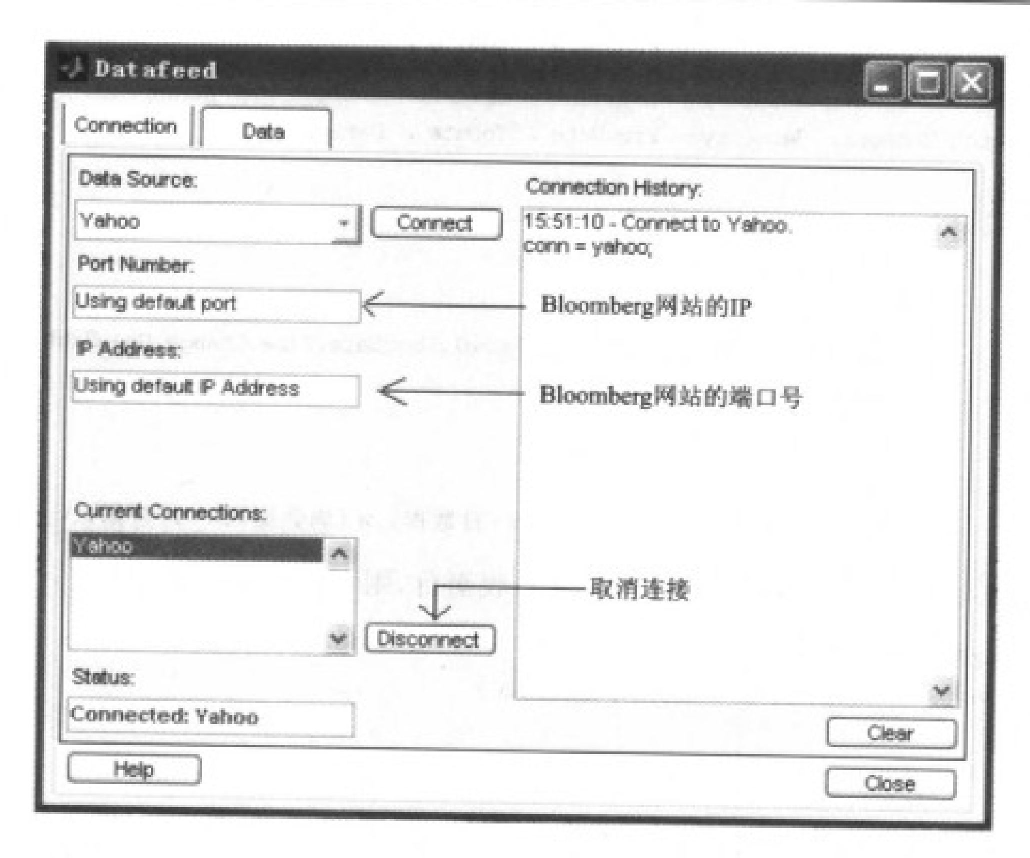


图 7.18　财经数据 GUI

在图 7.18 中，单击 Data 标签，自动弹出菜单如图 7.19 所示。

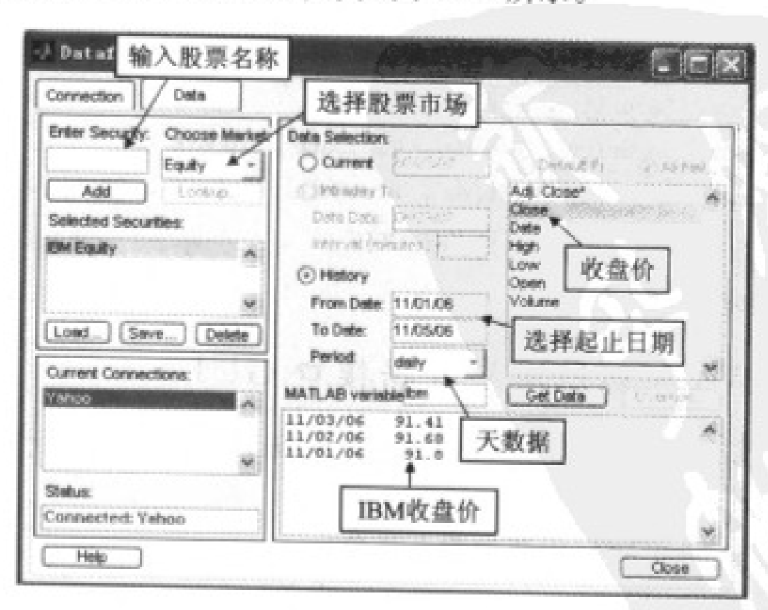


图 7.19　GUI 菜单内容

在Command窗口执行如下命令：

```
>> ibm
        732984      91.41
        732983      91.68
        732982      91.8
```

可以知道变量ibm中保存了IBM股票收盘价。

7.3 MATLAB和Word接口

Notebook具有Word全部功能，在M-Book模板中，文档、图像、表格及数学公式等的输入、排版、编辑与Word完全相同。

7.3.1 启动Notebook

1. 从MATLAB中启动Notebook

在MATLAB中直接执行命令启动。

```
>> notebook -setup
Welcome to the utility for setting up the MATLAB Notebook
for interfacing MATLAB to Microsoft Word
Setup complete
```

上述信息表明Notebook安装已经完成。

```
>> notebook
Warning: MATLAB is now an automation server
```

MTALAB提示已经启动Notebook，并且出现图7.20的界面。

2. 从Word中启动Notebook

方法1：打开MATLAB71\notebook\pc文件夹下M-book文件，直接进入编辑状态。

方法2：在Word菜单中选中【工具】菜单下的【模板和加载项】菜单项，然后打开【模板和加载项】对话框菜单，单击【选用】按钮，在弹出的对话框中选中M-book即可。



图 7.20　Notebook 界面

7.3.2　创建和运行 Word 中的计算区

计算区通常由 Word 文本和细胞(群)混合的若干段落构成,它被当作 Word 的一个“节”处理。MATLAB 提供计算区功能的目的在于使得计算区计算操作与电子文档其他部分独立,不管计算区中包含多少细胞群,只需一次操作就可以对所有细胞群进行计算。

计算区操作独立于 Word 文档其他部分,但是所有计算都是通过同一个 MATLAB“引擎”实现的,生成变量放在 MATLAB 同一个工作区中。

计算区可用于实时运行电子演讲稿中的 MATLAB 指令,而不用切换到 MATLAB 状态。

7.4 MATLAB与ActiveX接口

7.4.1 ActiveX基本介绍

ActiveX是一种基于Microsoft Windows系统组件集成协议，通过ActiveX，开发商和终端客户就把来自不同商家的ActiveX组件集成在自己的应用程序中，这样可以提高开发效率，避免在一些低层次的程序上花费更多时间。

ActiveX是面向对象技术集合，这些技术都有共同基础是“组件对象模型”COM(Component Object Model)。

MATLAB支持两种ActiveX，分别是ActiveX控件封装集成和ActiveX自动化。ActiveX控件(Control)是指那些可视、能编程的集成于ActiveX容器(containor)的应用组件，最常见是Microsoft Internet Explorer和Web Browser Control。

ActiveX自动化(automation)使得MATLAB控制、受控于其他软件。当MATLAB受控于其他软件时，MATLAB就变成了自动化服务器(automation server)；当MATLAB控制其他组件时，就表现为自动化客户(automation client)。MATLAB自动化服务器包括在MATLAB中执行指令、与MATLAB交换数据。

MATLAB自动化客户功能仅是MATLAB ActiveX控件封装集成功能的子集，所有ActiveX控件都是ActiveX自动化服务器，其内容如表7.3所列。

表7.3 Active控件

函数	功能
activecontrol	创建ActiveX控件对象，建立客户自动化支持
activeserver	创建ActiveX自动化服务器
Delete	删除ActiveX对象
Get	从接口获得属性值
Invoke	激活(或显示)接口
Move	在父窗口上移动对象或改变其大小
Propedit	要求控件显示其内建属性页
release	释放接口
send	返回事件列表
set	设置接口属性

【例7-2】 运行自动化服务器基本命令。打开Excel默认界面，增添工作簿(workbook)，激活当前页(worksheet)，实现MATLAB与Excel之间数据传输，保存Excel数据。

```
% 打开一个 Excel 服务器
>> Excel = actxserver('Excel.Application');
>> set(Excel, 'Visible', 1);                    % 开启页面可见
% 加入一个新工作簿句柄
>> Workbooks = Excel.Workbooks;
>> Workbook = invoke(Workbooks, 'Add');         % 产生新空白工作簿
% 激活第二张表
>> Sheets = Excel.ActiveWorkBook.Sheets;
>> sheet2 = get(Sheets, 'Item', 2);
>> invoke(sheet2, 'Activate');                  % 激活第二张表
% 激活工作表服务器
>> Activesheet = Excel.Activesheet;
% 把 MATLAB 中数据输入到 Excel 中
>> A = [1 2 4; 3 4 7];
% 矩阵 A 为 2 行 3 列,所以在 Excel 中"A1:C2"工作区可以保存数据
>> ActivesheetRange = get(Activesheet,'Range','A1:C2');
>> set(ActivesheetRange, 'Value', A);
% 取得一个区间,用于保存数据
>> Range = get(Activesheet, 'Range', 'A1:C2');
>> B = Range.value;
% 转换为双精度数据
>> B = reshape([B{:}], size(B));
% 保存工作簿
>> invoke(Workbook, 'SaveAs', 'myfile.xls');  % 保存到新工作簿
>> Workbook.Saved = 1;
>> invoke(Workbook, 'Close');
% 退出 Excel
>> invoke(Excel, 'Quit');
% delete(Excel);                                 % 结束整个过程
```

【例 7-3】 用二叉树方法计算欧式看涨期权,然后输出到 Excel 中。

```
% function binomial( )
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   利用 MATLAB 二叉树计算欧式看涨期权并将结果输出到 Excel 中              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s0 = 52;k = 50;r = 0.1;T = 5/12;dt = 1/12;sigma = 0.5;class = 1;
bb = [s0,k,r,T,dt,sigma];bb = bb';
```

```
[Price, Option] = binprice(s0, k, r, T, dt, sigma,class)
aa = cell(6,1);                    %定义单元数组
aa(1) = {'股价'};aa(2) = {'执行价'};aa(3) = {'无风险利率'};
aa(4) = {'波动率'};aa(5) = {'续存期'};aa(6) = {'时间步长'};
a1 = '欧式看涨期权的参数'
a2 = '欧式看涨期权的二叉树结构'
a3 = '欧式看涨期权的在二叉树节点的价格'
%%%%%%%%%%%%%%%%%%%%%%%%
%          激活 Excel 服务器                 %
%%%%%%%%%%%%%%%%%%%%%%%%
Excel  = actxserver('Excel.Application');
set(Excel, 'Visible', 1);
Workbooks  = Excel.Workbooks;
Workbook  =  invoke(Workbooks,'open', 'C:\Book1.xls');        %打开 Excel 文件。
Sheets  =  Excel.ActiveWorkBook.Sheets;
sheet1  = get(Sheets, 'Item', 1);
invoke(sheet1, 'Activate');      %激活第一张表
Activesheet  =  Excel.Activesheet;
%%%%%%%%%%%%%%%%%%%%%%%%
%    把 MATLAB 的数据传递给 Excel                %
%%%%%%%%%%%%%%%%%%%%%%%%
ActivesheetRange = get(Activesheet,'Range','A1');
set(ActivesheetRange, 'Value', a1);
ActivesheetRange = get(Activesheet,'Range','A3:A8');
set(ActivesheetRange, 'Value', aa);
ActivesheetRange = get(Activesheet,'Range','B3:B8');
set(ActivesheetRange, 'Value', bb);
ActivesheetRange = get(Activesheet,'Range','F1');
set(ActivesheetRange, 'Value', a2);
ActivesheetRange = get(Activesheet,'Range','D2:I7');
set(ActivesheetRange,'Value',Price);
ActivesheetRange = get(Activesheet,'Range','E10');
set(ActivesheetRange,'Value',a3);
ActivesheetRange = get(Activesheet,'Range','D11:I16');
set(ActivesheetRange,'Value',Option);
%保存工作簿
invoke(Workbook, 'Save');
```

```
invoke(Excel, 'Quit');
delete(Excel); %                     结束整个过程
```

运行结果如图 7.21 所示。

	A	B	C	D	E	F	G	H
1	欧式看涨期权的参数				欧式看涨期权的二叉树结构			
2			52.00	60.07	69.40	80.18	92.63	107.01
3	股价	52.00	0.00	45.01	52.00	60.07	69.40	80.18
4	执行价	50.00	0.00	0.00	38.96	45.01	52.00	60.07
5	无风险	0.10	0.00	0.00	0.00	33.72	38.96	45.01
6	波动率	0.42	0.00	0.00	0.00	0.00	29.19	33.72
7	续存期	0.08	0.00	0.00	0.00	0.00	0.00	25.27
8	时间步	0.50						
9								
10				欧式看涨期权的在二叉树节点的价格				
11			8.90	14.00	21.27	31.01	43.04	57.01
12			0.00	4.09	7.16	12.16	19.82	30.18
13			0.00	0.00	1.18	2.41	4.92	10.07
14			0.00	0.00	0.00	0.00	0.00	0.00
15			0.00	0.00	0.00	0.00	0.00	0.00
16			0.00	0.00	0.00	0.00	0.00	0.00

图7.21 利用 ActiveX 将欧式期权二叉树输出到 Excel 中

7.4.2 MATLAB ActiveX 自动化服务器

通过 MATLAB ActiveX 自动化服务器，用户可以在自己的应用程序中执行 MATLAB 命令，并可以与 MATLAB 工作区间进行数据连接，如果将 MATLAB 作为服务器使用，则首先要查阅希望使用自动化服务器的应用程序文档，了解如何在控制器中开启自动化服务器。MATLAB ActiveX 对象在系统注册表中定义名称为 ProgID，通常 ProgID 自动化服务器取下面两个名字之一，MATLAB. Application（将启动的 MATLAB 自动化服务器作为共享服务器）或 MATLAB. Application. Single（将启动的 MATLAB 自动化服务器作为专用服务器）。

思考题

1. 将金牛股份(000937)2005 年 10 月 14 日至 2006 年 1 月 10 日的股价数据保存在 Excel 中，然后通过 Excllink 将收盘价保存到 MATLAB 中。计算收益率的均值、方差和夏普比率，然后传回 Excel 中。

2. 将思考题 1 的步骤用 MATLAB 程序完成。

第 8 章　MATLAB 与 VBA 混合编程

通过 VBA 可以实现对 Excel 的操作，加强了 Excel 的功能，把 Excel 的操作放在后台进行，使得 Excel 的前台非常简洁。本章要求读者掌握 VBA 的基本函数，掌握使用 VBA 操作 MATLAB 的常用方法。

8.1　VBA 基础知识

8.1.1　VBA 基本介绍

VBA(Visual Basic For Applications)是 Microsoft 公司开发出来在 Windows 桌面应用程序中执行通用自动化任务(OLE)的编程语言。VBA 是 Visual Basic 的一个子集。VBA 与 VB 的区别在于 VBA 要求有一个宿主应用程序才能运行，而且不能用于创建的独立的应用程序，而 VB 可用于创建独立的应用程序。VBA 继承了 VB 很大一部分的编程方法，VB 的语法结构、变量的声明以及函数的调用和 VBA 一样。

VBA 是面向对象的编程语言，每一个对象代表 Excel 中的一个部件或者一项功能，例如工作簿、工作表、单元格、区域、图表及情节等都可以作为 Excel 的对象，Excel 本身也可以作为一个对象让 VBA 进行操作。

对象具有下面 4 个特征：

- 对象是一个集合。
  例如 Workbooks 对象是所有已经打开的工作簿的集合，Worksheets 也是所有工作表的集合。有些对象只有一个成员，例如 Application、单元格。这些单一的对象可以直接被引用，简单写为 Application.、Font.（在对象后面加一个后缀.）。引用时 Worksheet(1)是指第一个工作表，Worksheet("inputs")表示引用工作表名为“inputs”的工作表。
- 对象具有层次。
  如果引用的是名称，则名称加上双引号，也可以是按照位置或者序号引用。例如 Application. Workbooks("Model. xls"). sheets("inputs"). range("data")。该引用中工作簿是 Model. xls，工作表的名称是 inputs，单元格区域是 data。
- 对象具有属性。
  属性是描述对象的特征，属性值一般为数字、文本、True 或者 False 等。例如 Applica-

tion. ScreenUpdating＝False，ScreenUpdating 是 Application 的属性，表示是否运行时刷新屏幕，"False"表示关闭运行时刷新屏幕的功能。

● 对象具有方法。

方法是对对象实施的一系列操作。例如

```
Range("A1:B3").Select              表示选中单元格区域 A1:B3
Range("A1:B10").Copy               表示复制单元格区域 A1:B10 的内容
Range("A11:B11").PasteSpecial      粘贴剪切板中内容到单元格区域 A11:B11
```

上面例子中 Range 是对象，"Select"，"Copy"，"PasteSpecial"都属于方法。对象 Workbook 和 Worksheet 也有相应的方法。

```
Workbooks("Model.xls").Activate      激活名称为 Model.xls 的工作簿
Sheets("inputs").Delete              删除名称为 inputs 的工作表
```

8.1.2 VBA 编辑窗口的结构

打开 VBA 的方法有很多。最常用的方法是按 Alt＋F11 组合键，此外也可以在 Excel 中单击 "Visual Basic"菜单的【宏】中的【Visual Basic 编辑器】子菜单项，如图 8.1 所示。

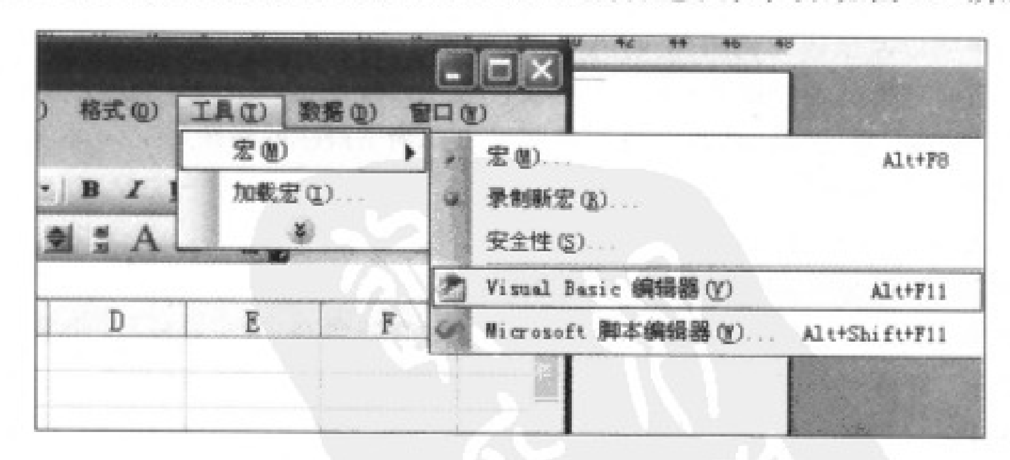


图 8.1 Excel 中工具栏 VBA 编辑器

打开后的 VBA 编辑器界面如图 8.2 所示。

从图 8.2 可以看出，VBA 由下面几个部分构成：

● 菜单及工具栏。

● 工程窗口。

在 VBA 的工程窗口中每个打开的工作簿就是一个工程，工程的默认名为"VBAProject(工作簿名称)"；工程还包括窗体、模块和类模块。窗体用于显示应用程序的对话框。模块则是代码的集合，是由变量声明、过程构成。过程是一个可以单独执行的

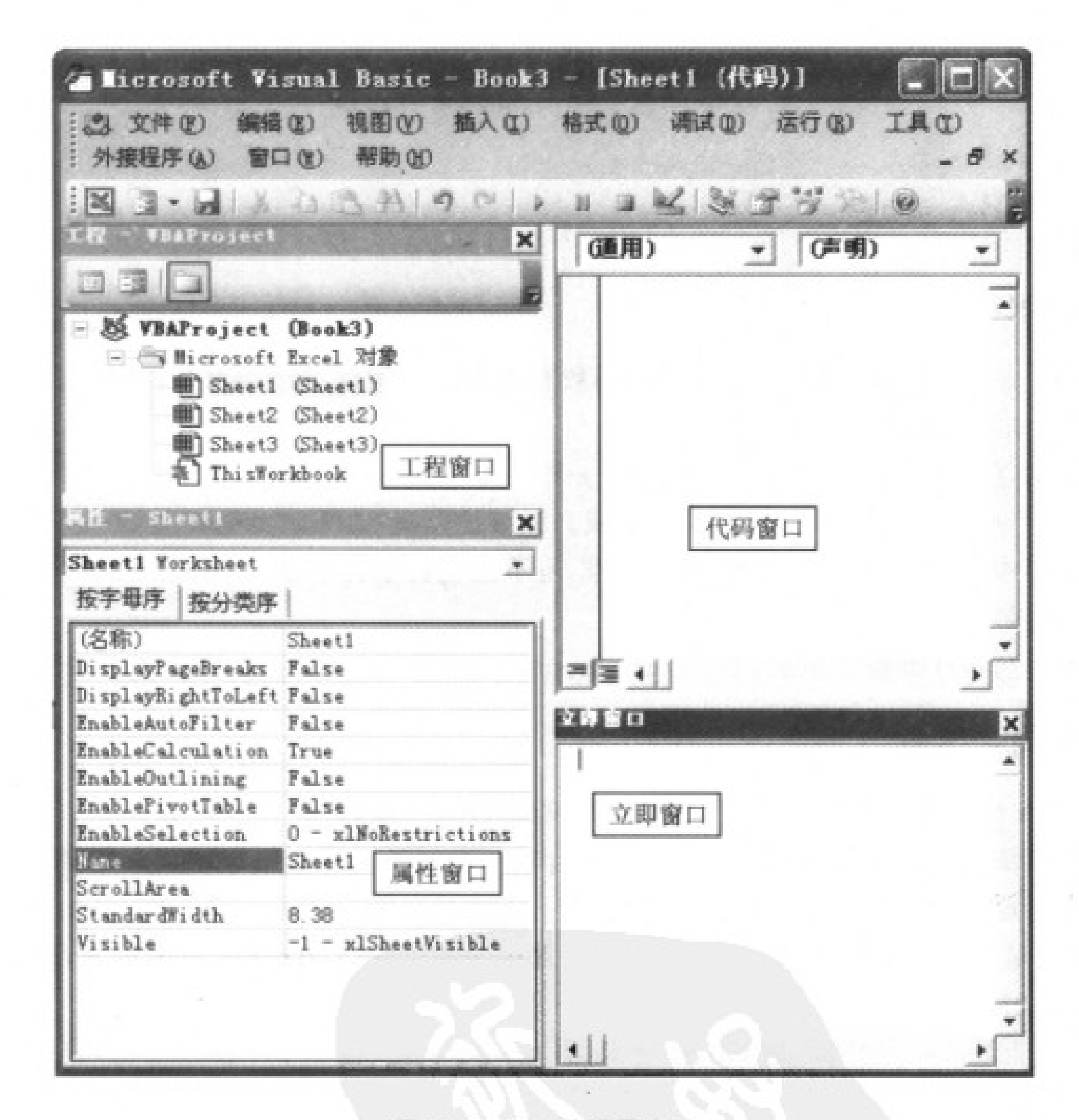


图 8.2　VBA 编辑器窗口

代码语句，模块则是过程的集合。类模块允许开发人员自己开发对象。

- 属性窗口。

  属性窗口显示对象以及窗体的属性。

- 代码窗口。

  代码列表用于编写和保存 VBA 代码。

- 立即窗口。

  显示运行结果。

8.2 VBA编程指南

8.2.1 VBA 变量

1. 标识符

标识符是一种标识变量、常量、过程、函数和类等语言构成单位的符号，利用它可以完成对变量、常量、过程、函数和类等的引用。

VBA 对子程序和函数的命名规则如下：

- 名字中可以包含字母、数字和下划线。
- 名字中不能包含空格、句号和惊叹号，也不能包含字符@、&、$及#。
- 名字最多可以包含 255 个字符。
- 不能与 VB 保留字重名，如 public，private，dim，goto，next，with，integer，single 等。

2. 变量与常量

① VBA 允许使用未定义的变量，默认是变体变量 Variant。

② 在模块通用说明部分，加入 Option Explicit 语句可以强制用户进行变量定义。

③ 变量定义语句及变量作用域。

一般变量作用域的原则是，在哪部分定义就在哪部分起作用；在模块中定义，则在该模块那部分起作用。

常量为变量的一种特例，用 Const 定义，且定义时赋值，程序中不能改变值，作用域也如同变量作用域。例如：Const Pi=3.1415926 as single。

3. 数　组

数组是包含相同数据类型的一组变量的集合，对数组中的单个变量引用通过数组索引下标进行。在内存中表现为一个连续的内存块，必须用 Global 或 Dim 语句来定义。定义规则如下：Dim 数组名([lower to]upper [，[lower to]upper，….]) as type ；Lower 的默认值为 0。二维数组是按行列排列，如 XYZ(行，列)。

除了以上固定数组外，VBA 还有一种功能强大的动态数组，定义时无大小维数声明；在程序中再利用 Redim 语句来重新改变数组大小，原来数组内容可以通过加 preserve 关键字来保留，如 Dim array1() as double ：Redim array1(5) ：array1(3)=250 ：Redim preserve array1(5,10)。

VBA 的变量数据类型如表 8.1 所列。

表8.1 VBA的变量数据类型

数据类型	类型标识符	字　节
字符串型(String)	$	字符长度(0～65 400)
字节型(Byte)	无	1
布尔型(Boolean)	无	2
整数型(Integer)	%	2
长整数型(Long)	&	4
单精度型(Single)	!	4
双精度型(Double)	#	8
日期型(Date)	无	8(公元100/1/1—1999/12/31)
货币型(Currency)	@	8
小数点型(Decimal)	无	14
变体型(Variant)	无	以上任意类型,可变
对象型(Object)	无	4

8.2.2 VBA运算符

在VBA编程时需要对变量进行计算,运算符是计算机语言最基本的内容。和其他软件一样,Excel VBA运算符的用法如表8.2所列。

表8.2 VBA常用的运算符列表

运算符	说　明	例　子	运算结果
+	加	3+3	6
−	减	3−3	0
*	乘	3*3	9
/	除	3/3	1
\	整除	10\4	2
^	乘方	3^2	9
=	赋值	X=10	
And	同时为真	x=4,y=4, x>1 And y<10	False
Or	或	x=4,y=4, x>1 Or y<10	True
Like	字符串比较	"ABCDEFG" Like "*DE*"	True
Mod	整除后余数	Mod(19,6)	1

续表 8.2

运算符	说　明	例　子	运算结果
Xor	两个条件中一项为真	myxor = myxor Xor range(i, j)	
not	否	IsError(x)=False , Not IsError(x)	True
&	字符连接	"North"&"wind"	Northwind

8.2.3 VBA常用属性

【例8-1】 新建一个Excel文件，打开VBA编辑器，设定单元格A1的字体为宋体，大小为14，颜色为红色，粗体。

```
Sub 字体格式()
Range("A1").Font.Name = "宋体"                '单元格A1字体为"宋体"
Range("A1").Font.Size  = 14                   '单元格A1字体大小为14
Range("A1").Font.ColorIndex = 3               '单元格A1字体颜色为红色
Range("A1").Font.Bold = True                  '单元格A1字体为粗体
Range("A1").value = "数据库"                  '单元格A1内容
End Sub
```

代码内容如图8.3所示。

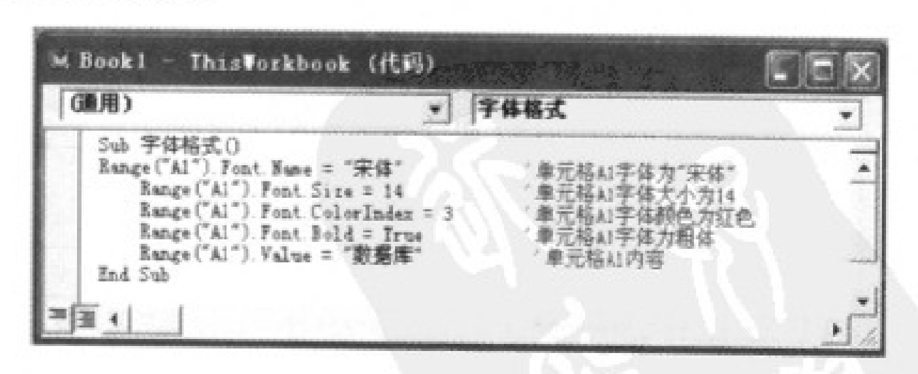


图8.3　给单元格A1赋值的VBA代码

在图8.3的界面上，用鼠标选择执行的代码，打开【运行】菜单，选择【运行子过程/运行窗体】菜单项，或者按【F5】键，发现A1单元格被赋予新的内容。

8.2.4 VBA的控制语句

1. With语句

【例8-2】 用With语句改写【例8-1】。

```
Sub 字体格式()
with range("A1").Font
    .name = "宋体"
    .Size  = 14
    .colorindex = 3
    .bold = true
End Sub
```

运行该子程序可以发现字体属性发生了变化。注意 with 语句中的属性前面要加上“.”号，表示是 range("A1").Font 的属性。

2. If…Then…Else 语句

【例 8-3】 判断单元格 A1 的元素是否大于 1。如果大于 1，则单元格 A2 内容为“是大于 1”；如果小于 1，则单元格 A2 内容为“小于 1”。

```
Sub 是否大于 1()
If  Range("A1").value>1 Then          'Range("A1").value 表示单元格 A1 的值
    Range("A2").value = "是大于 1"
Else
Range("A2").value = "小于 1"
End  If
End  Sub
```

3. Case 语句

【例 8-4】 单元格 A1 为股票的价格，现在判断如果股票价格大于 20 元，则 A2 内容为“高价股”，小于 5 元为“低价股”，其他为“中价股”。

```
Sub 判断股价类型()
Select Case  Range("A1")
Case Is >20
    Range("A2").value = "高价股"
Case Is <20
Range("A2").value = "低价股"
End  Select
End  Sub
```

4. For 循环语句

循环语句是比较常用的语句。

【例 8-5】 为单元格 A1 到 A5 依次赋值 1～5。

```
Sub 单元格赋值()
    Dim I as Integer          '定义变量 I 为整数型变量
    For I = 1 To 5 Step = 1   'I 为从 1～5,步长为 1
    Range("A"&I) = I
Next I
End Sub
```

8.2.5 VBA 的主要功能

Excel 内置有大约 330 个函数,这些函数可以分成 11 个大类:数据库函数、日期和时间函数、外部函数、工程函数、财务函数、信息函数、逻辑函数、查找和引用函数、数学和三角函数、统计函数以及文本和数据函数。下面介绍一些常用的函数。

1. 工作表的选取

【例 8-6】 选取特定工作表。

```
Sub 选取特定工作表 sheet2()
Sheets("sheet2").Select
End Sub
```

注意只有在工作表选定的情况下才能选定单元格。

2. 单元格的选取操作

选取单元格是操作单元格的前提,下面介绍选取单元格的几种方法。

【例 8-7】 选取单个单元格。

方法 1

```
Sub 选取单个单元格 B2()
Range("B2").Select
End Sub
```

方法 2

```
Sub 选取单个单元格 B2()
Range("B"&2).Select
End Sub
```

方法 3

```
Sub 选取单个单元格 B2()
Cells(2,2).Select
End Sub
```

方法 4

```
Sub 选取单个单元格 B2()
Range(2,"B").Select
End Sub
```

方法 5

```
Sub 选取单个单元格 B2()
[B2].Select
End Sub
```

【例 8-8】 选取特定的单元格区域 A1:B4。

方法 1

```
Sub 选取连续的单元格区域()
Range("A1:B4").Select
End Sub
```

方法 2

```
Sub 选取连续的单元格区域()
Range(Cells(1,1),Cells(4,2)).Select
End Sub
```

方法 3

```
Sub 选取连续的单元格区域()
[A1:B4].Select
End Sub
```

如果单元格是不连续的话，则可以采用下面的方法。

【例 8-9】 选取特定的不连续单元格区域 A1,C2:C5,E2:E5。

```
Sub ss()
Sheets(1).Select
Sheets(1).Range("A1,C2:C5,E2:E5").Select
End Sub
```

【例 8-10】 选取第 3 行、第 4 列。

选取第 3 行。

```
Sub 选取特定的行()
Rows(3).Select
Cloumns(4).Select
End Sub
```

选取第 4 列。

```
Sub 选取特定的列()
Sheets(1).Select
Columns(4).Select
End Sub
```

同时选定 2:3 行、4:5 列

```
Sub 选取特定的行列()
Sheets(1).Select
range("2:3,D:E").Select
End Sub
```

8.2.6 VBA 的查找功能

【例 8-11】 3 只股票的价格如表 8.3 所列。在 Excel 中录入数据并查找 2007 年 8 月 23 日深发展的股票价格。结果保存在单元格 D8 中。

表 8.3 上证指数、深发展及浦发银行股价图

项目 日期	上证指数	深发展	浦发银行
2007-8-20	4904.852	37.9	44.96
2007-8-21	4955.211	38.56	46.9
2007-8-22	4980.082	38.59	49.39
2007-8-23	5032.491	38.39	49.95
2007-8-24	5107.67	39.6	52.68

将表 8-1 的内容保存在 Book1.xls 文件中，如图 8.4 所示。

下面是程序。

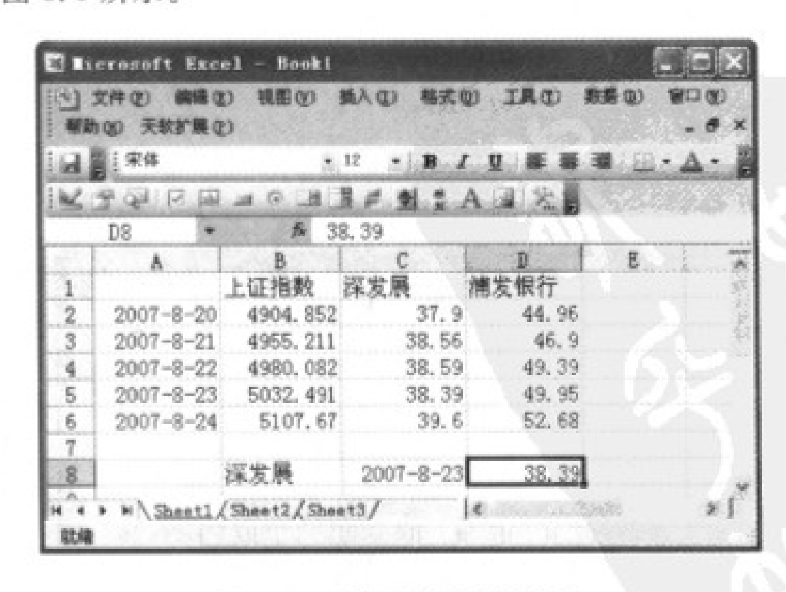
```
Sub 查找股票价格( )
Dim X As Integer
```

```
Dim Y As Integer
Y = Sheets(1).Range("C1:C4").Find(Range("B8")).Column
X = Sheets(1).Range("A1:A6").Find(Range("C8")).Row
Range("D8") = Cells(X, Y)          '将查找的结果保存到 D8 中
End Sub
```

	A	B	C	D	E
1		上证指数	深发展	浦发银行	
2	2007-8-20	4904.852	37.9	44.96	
3	2007-8-21	4955.211	38.56	46.9	
4	2007-8-22	4980.082	38.59	49.39	
5	2007-8-23	5032.491	38.39	49.95	
6	2007-8-24	5107.67	39.6	52.68	
7					
8		深发展	2007-8-23		

图 8.4　Book1.xls 文件内容

查询的结果如图 8.5 所示。



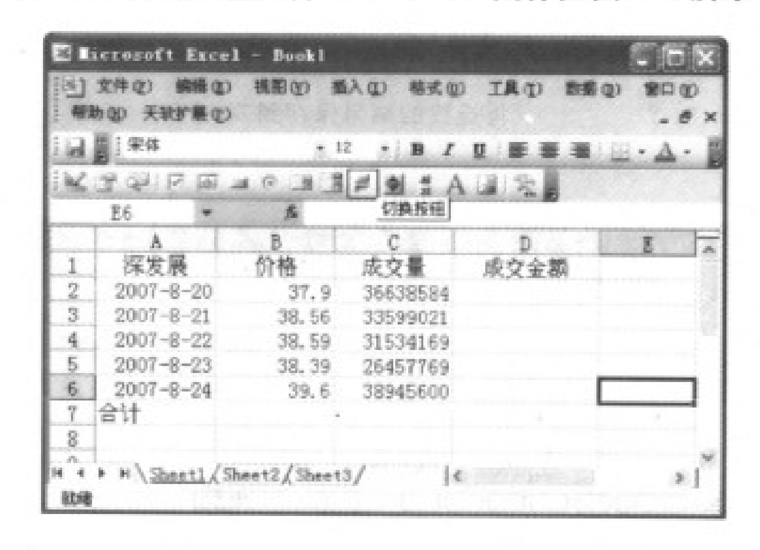
	A	B	C	D	E
1		上证指数	深发展	浦发银行	
2	2007-8-20	4904.852	37.9	44.96	
3	2007-8-21	4955.211	38.56	46.9	
4	2007-8-22	4980.082	38.59	49.39	
5	2007-8-23	5032.491	38.39	49.95	
6	2007-8-24	5107.67	39.6	52.68	
7					
8		深发展	2007-8-23	38.39	

图 8.5　查询日期对应的股价

8.2.7　VBA 的计算

【例 8-12】　深发展的股价和成交量如表 8.3 所列，试计算每天成交金额，并汇总。

首先建立深发展的股价和成交量文件 Book1. xls，具体如图 8.6 所示。



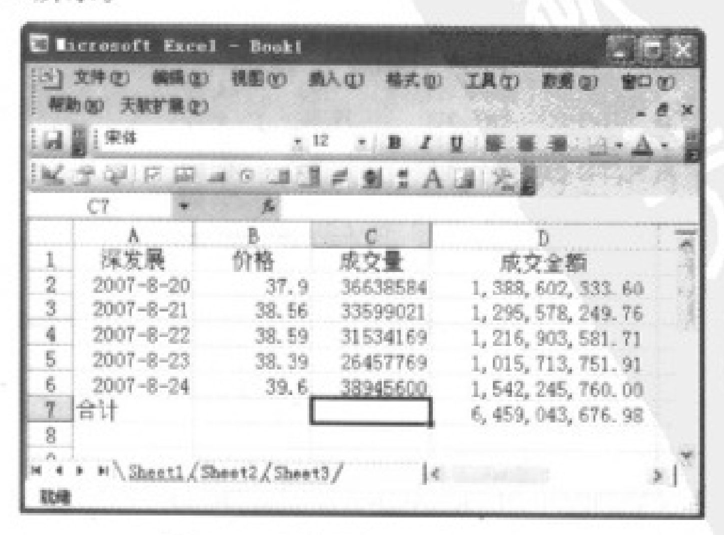
	A	B	C	D	E
1	深发展	价格	成交量	成交金额	
2	2007-8-20	37.9	36638584		
3	2007-8-21	38.56	33599021		
4	2007-8-22	38.59	31534169		
5	2007-8-23	38.39	26457769		
6	2007-8-24	39.6	38945600		
7	合计				

图 8.6 深发展的股价成交量的 Excel 文件

编写深发展每天的成交金额和汇总的公式如下：

```
Sub 深发展的成交金额汇总()
Dim i As Integer
For i = 2 To 6
Range("D" & i).Value = Range("B" & i).Value * Range("C" & i).Value
Next i
Range("D" & 7) = Application.Sum(Range("D2:D6"))
End Sub
```

运行结果如图 8.7 所示。



	A	B	C	D
1	深发展	价格	成交量	成交金额
2	2007-8-20	37.9	36638584	1,388,602,333.60
3	2007-8-21	38.56	33599021	1,295,578,249.76
4	2007-8-22	38.59	31534169	1,216,903,581.71
5	2007-8-23	38.39	26457769	1,015,713,751.91
6	2007-8-24	39.6	38945600	1,542,245,760.00
7	合计			6,459,043,676.98

图 8.7 深发展交易金额汇总图

8.2.8 VBA的窗体

下面将【例8-12】的函数通过窗体按钮进行操作。具体步骤如下：

在Excel菜单【视图】中的【工具栏】下选择的【窗体】子菜单项，具体如图8.8所示。

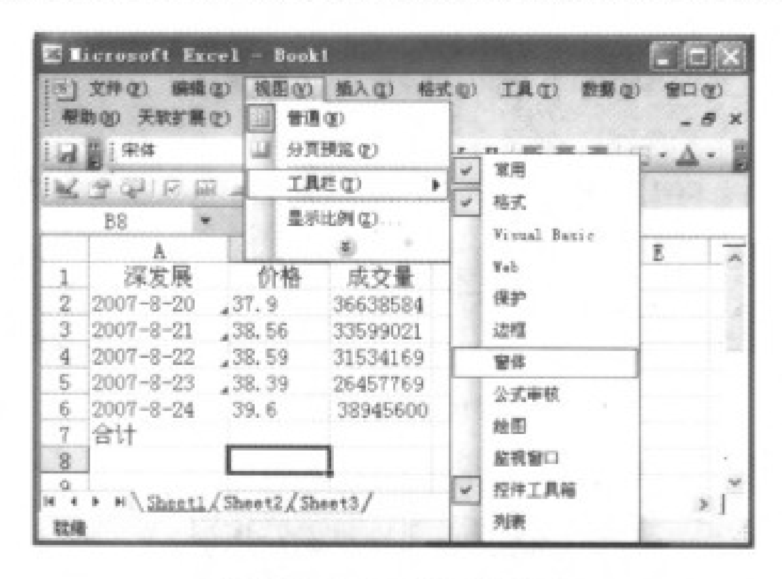


图8.8 Excel中的视图窗口

这时发现【窗体】工具条出现在Excel工作簿中，具体如图8.9所示。

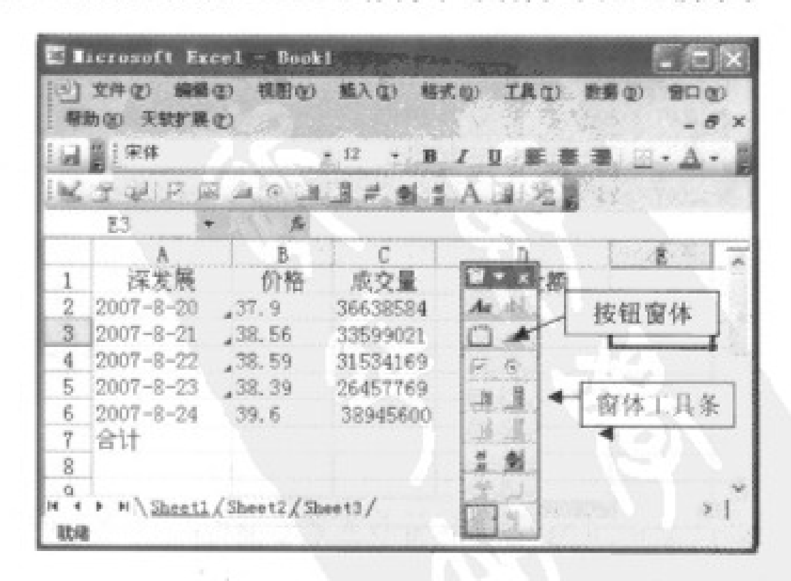


图8.9 Excel中的窗体工具条

在图8.9中单击按钮窗体的图标，将光标放在合适位置，然后拖动鼠标建立“按钮1”，具体如图8.10所示。

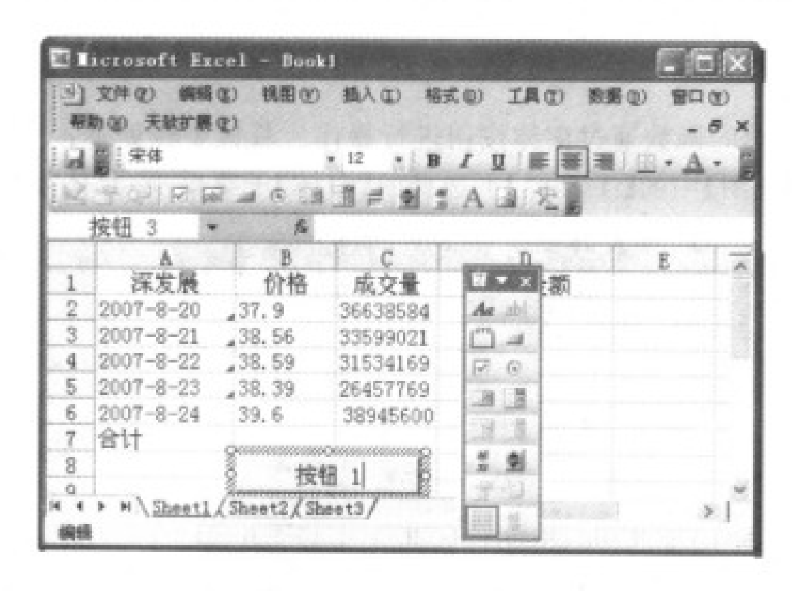


图 8.10 建立“按钮 1”

将光标放在“按钮 1”上，右击出现选项菜单，具体如图 8.11 所示。

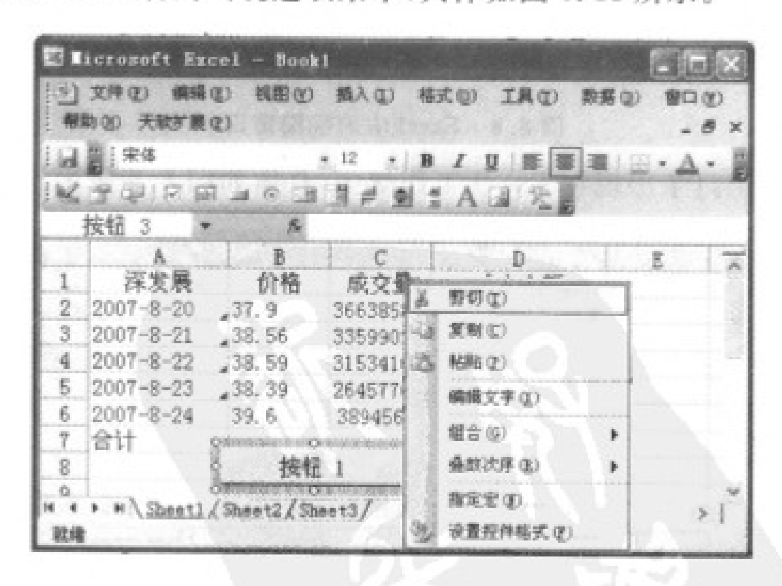


图 8.11 按钮工具条选项

在图 8.11 的窗口中，选择【指定宏】菜单项，就会出现如图 8.12 所示的对话框。

在图 8.12 的对话框中为按钮指定“深发展的成交金额汇总”函数，单击【确定】按钮退出对话框。然后在图 8.11 的窗口中，选择【编辑文字】菜单项，将“按钮 1”的名称换为“深发展的成交金额汇总”，具体如图 8.13 所示。

在图 8.13 中单击【深发展的成交金额汇总】按钮执行“深发展的成交金额汇总”函数，深发展的成交金额显示在相应的表格中，具体如图 8.14 所示。



图 8.12 【指定宏】对话框　　　　图 8.13 完成按钮设置后的界面

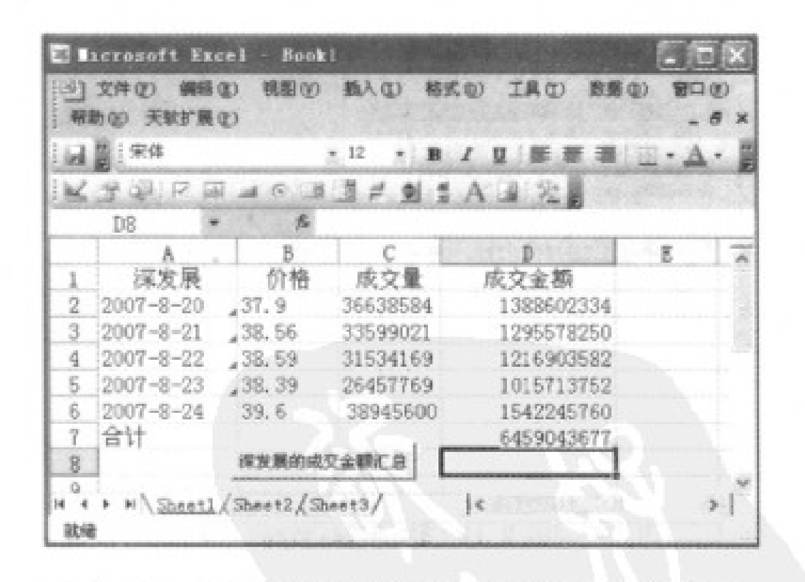


图 8.14 执行"深发展的成交金额汇总"函数后的界面

8.3 MATLAB 和 VBA 混合编程

8.3.1 建立和 Excel 的连接

在 VBA 中调用 MATLAB 函数需要引用 excllink 文件。引用的过程如下：

编辑窗口中单击【工具】菜单，出现的菜单如图 8.15 所示。

在图 8.15 的菜单中选择【引用】菜单项，出现图 8.16 的对话框。



图 8.15 VBA 编辑窗口的引用选项

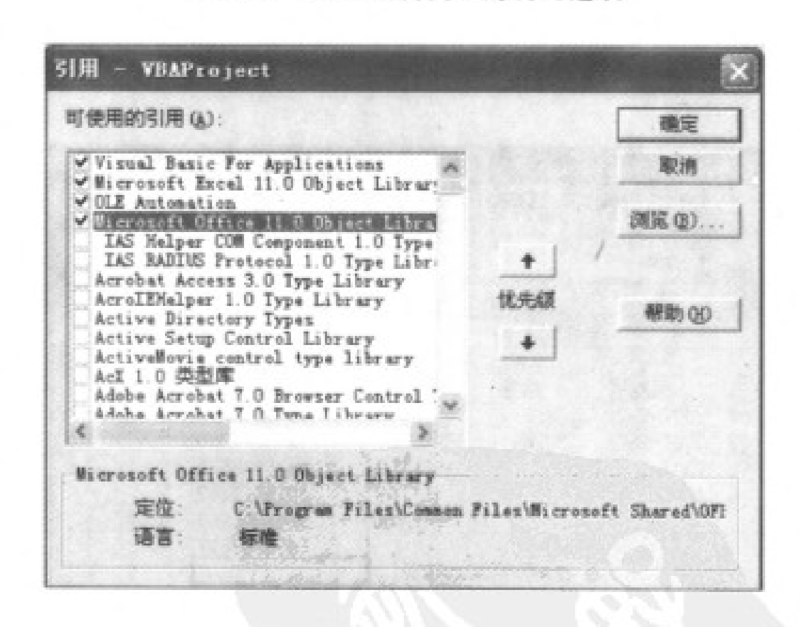


图 8.16 【引用】对话框

在图 8.16 的对话框中单击【浏览】按钮，弹出如图 8.17 所示的对话框。

在图 8.17 的【浏览】对话框中，找到“toolbox\exlink”目录下的 excllink 文件，单击【打开】按钮。则可以看到【excllink】前打了钩，说明 excllink 被成功引用了。具体如图 8.18 所示。

在图 8.18 窗口中单击【确定】按钮退出对话框，回到 VBA 编辑窗口可以看到目录树中“引用 excllink”成功。具体如图 8.19 所示。



图 8.17　【浏览】对话框

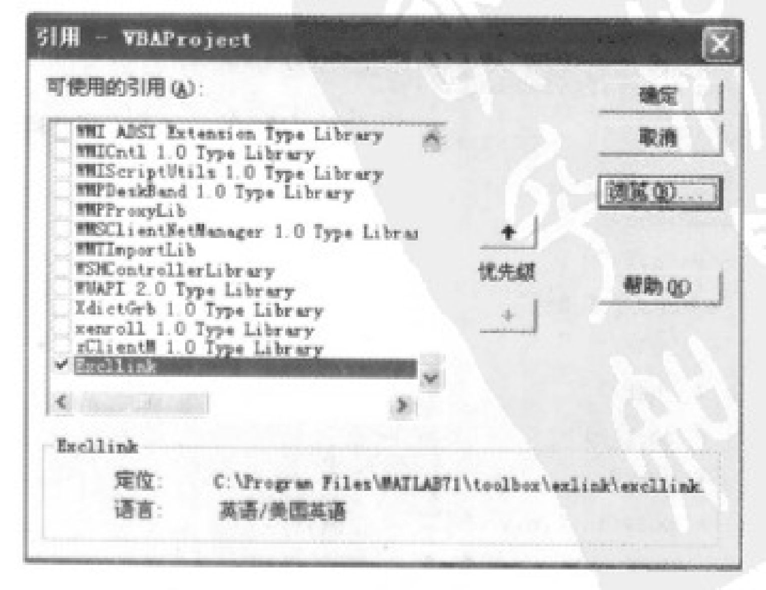


图 8.18　Excllink 文件引用标志

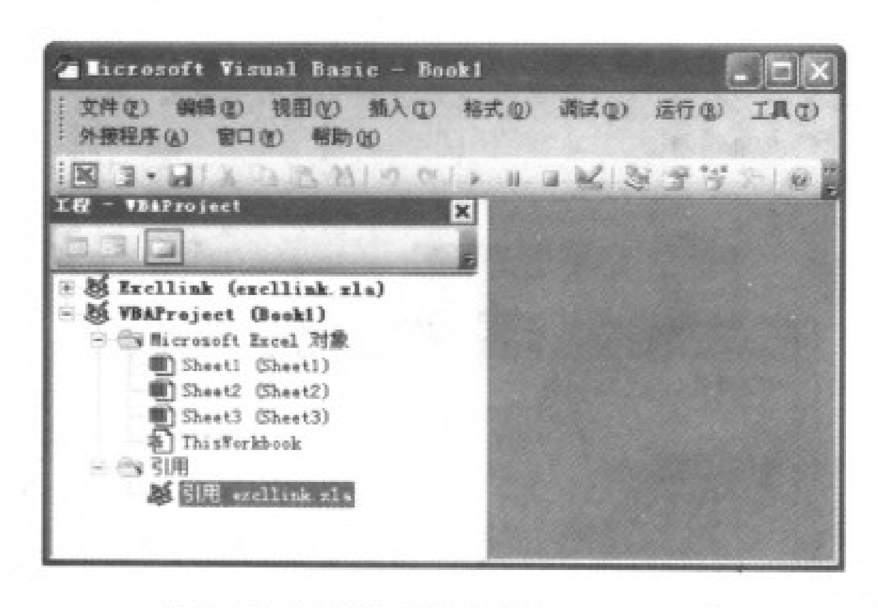


图 8.19 VBA 编辑窗口的"引用 excllink.xla"

8.3.2 MATLAB 与 VBA 混合编程

excllink 函数提供了操作 MATLAB 的 VBA 命令，其内容如表 7.2 所列。下面利用 MATLAB 自带的文件说明 VBA 操作 MATLAB 的方法。

打开"MATLAB 71\toolbox"目录下的 ExliSamp.xls 文件，进入 VBA 编辑窗口，选择【模块】后可以看到 VBA 代码。具体如图 8.20 所示。

模块的功能是进行数据拟合，然后进行绘图。

VBA 中代码的内容如下：

```
Function CurveFit(aData, sTarget1, sTarget2, sTarget3)
'MATLAB regression and curve fitting macro
MLPutMatrix "data", aData                          '将变量 aData 内容传给 MATLAB 中变量 data
MLEvalString "y = data(:,3)"                       '执行 MATLAB 命令
MLEvalString "n = length(y)"
MLEvalString "e = ones(n,1)"
MLEvalString "A = [e data(:,1:2)]"
MLEvalString "beta = A\y"
MLEvalString "fit = A * beta"
MLEvalString "[y,k] = sort(y)"
MLEvalString "fit = fit(k)"
MLEvalString "[p,S] = polyfit(1:n,y',5)"         '拟合
MLEvalString "newfit = polyval(p,1:n,S)'"
MLEvalString "plot(1:n,y,'bo',1:n,fit,'r:',1:n,newfit,'g')"
```

```
MLEvalString "legend('data','fit','newfit')"
MLGetMatrix "y", sTarget1                    '将 MATLAB 中 y 值赋给 sTarget1
MLGetMatrix "fit", sTarget2
MLGetMatrix "newfit", sTarget3
End Function
```

注意和 MATLAB 有关的内容都要加双引号。

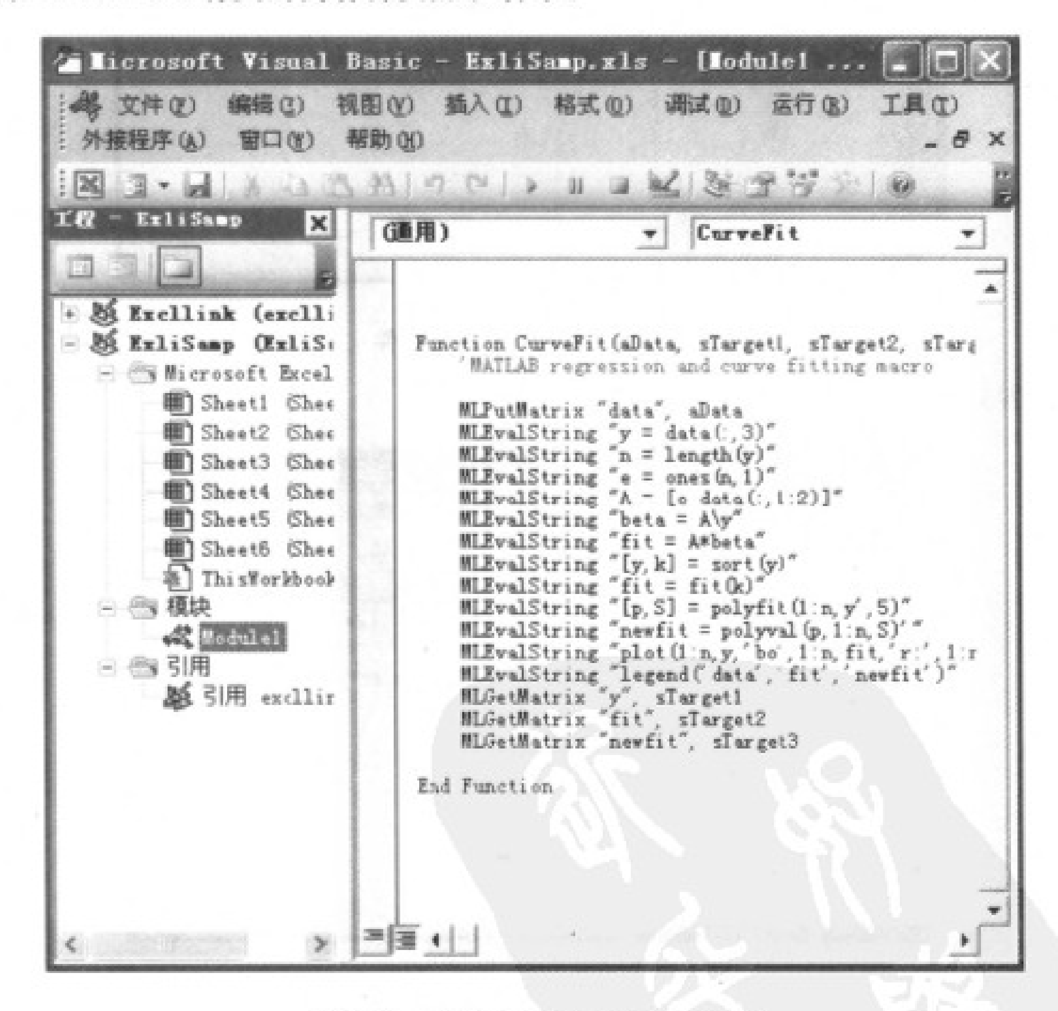


图 8.20　ExliSamp.xls 中"模块"代码

【例 8-13】 将 Excel 文件中单元格区域导入 MATLAB 中。

首先在 C:\Documents and Settings\ader\My Documents 目录下建立一个名为 Book1 的文件，内容如图 8.21 所示，并将表 Sheet1 的单元格区域"A1:B2"命名为 disc。

然后在 VBA 中将第一张表的单元格区域 A1:B2 命名为 disc。代码如下：

```
Sub sd( )
Sheets(1).Range("A1:B2").Name = "disc"          '将第一张表的 A1:B2 区域命名为 disc
```

```
End Sub
```

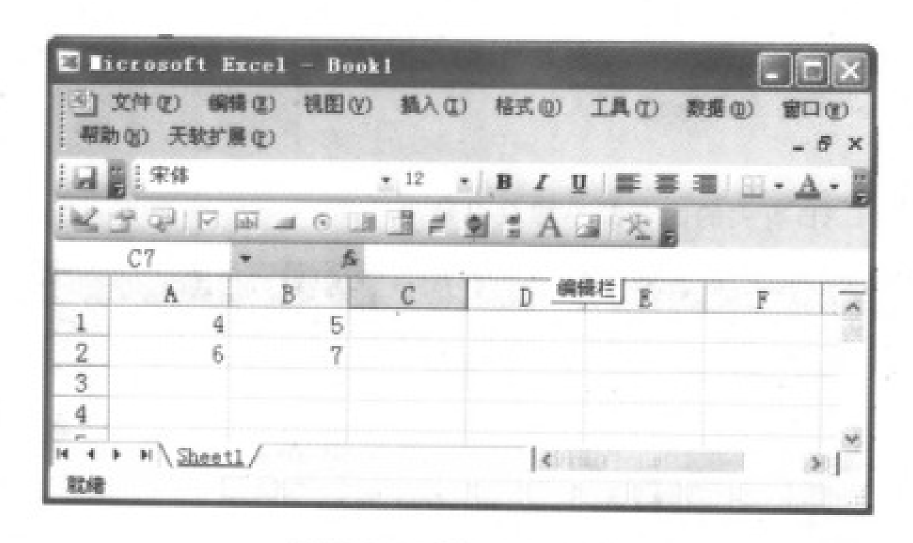


图 8.21 建立 Book1 文件

具体内容如图 8.22 所示。

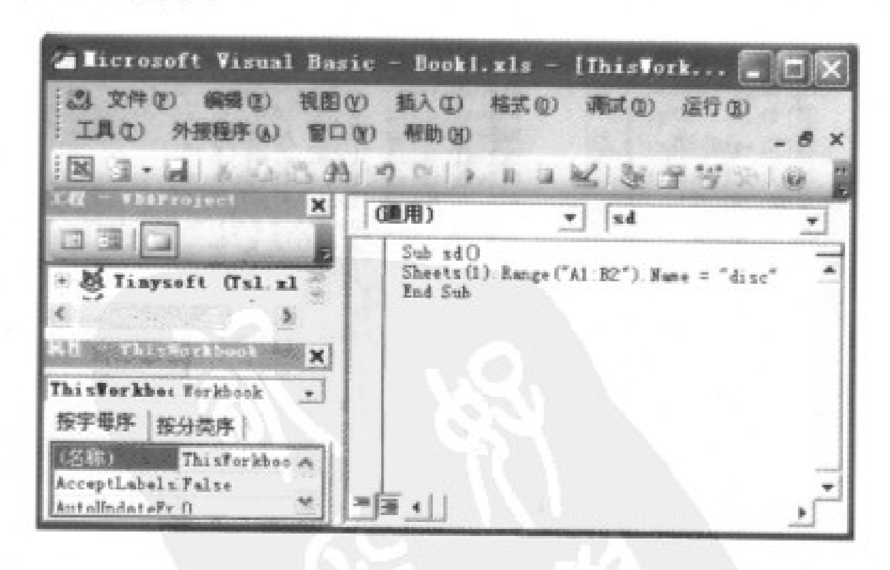


图 8.22 命名单元格区域

在图 8.22 中选中子程序代码并执行，具体如图 8.23 所示。

然后保存、退出该文件。

从 MathWorks 公司的文件交换网站上下载一个 m 文件，源程序从 MathWorks 公司文件交换网站下载：http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=16047&objectType=file，文件名为 Store Excel named range in MATLAB。

该文件可以获取 disc 的内容，文件代码如下：

```
function MATLABValue = GetNamedRangeFromExcel(RangeName, ExcelFile)
```

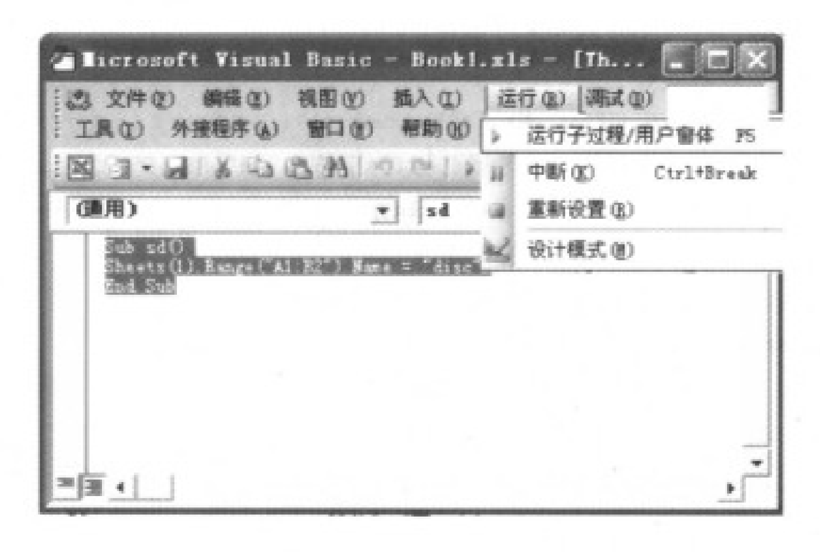


图8.23　运行选中的子程序

```
%将Excel中文件单元格区域内容导入到MATLAB中
%输入变量
%        RangeName          单元格区域名
%        ExcelFile          Excel文件路径
%   输出变量
%        MATLABValue        单元格区域内容
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 建立和Excel的数据连接
Excel = actxserver( 'excel.application' );
Excel.Visible = 0;
%   打开Excel文件ExcelFile
workbook = Excel.workbooks.Open( ExcelFile );
%   取得ExcelFile中的所有工作表名称
NamedRangeCount = Excel.ActiveWorkBook.names.count;
% 查找单元格名称RangeName
if NamedRangeCount > 0
    for ii = 1:NamedRangeCount
        RName = get( Excel.ActiveWorkbook.names.Item( ii ) );
        % 判断是否为RangeName
        if ( strcmp(RName.Name, RangeName) )
          % 如果找到的话执行下面操作
          RangeOfNamedCells   = RName.Value;       %其形式如"=Sheet1! &A&1:&B&2"
          RangeOfNamedCells(1) = [];               %去掉"="
          %看看那个工作表中含有RangeName
```

```
            pos = findstr(RangeOfNamedCells,'!');
            % 获得工作表名称
            SheetName = RangeOfNamedCells(1: pos - 1);
            % RangeName 的位置单元格区域名称，为字符串，如"&A&1:&B&2"
            NameOfTheRange = RangeOfNamedCells(pos + 1 : end);
            % 继续寻找 RangeName 所在的工作表
            xlSheet = workbook.get('Sheets',SheetName);
            % 建立连接，得到单元格区域的内容
            eRange = xlSheet.get('Range', NameOfTheRange);
            MATLABValue    = eRange.Value;
             % 退出循环
            break;
            end
          end
end
if ( ~exist('MATLABValue','var') )
    % 如果单元区域没有找到返回空变量
    warning('文件中没有目标单元区域变量.');
    MATLABValue = [];
end;
% 关闭和 Excel 的链接
Excel.Quit;
Excel.delete;
```

下面在 MATLAB 命令窗口下运行 GetNamedRangeFromExcel 程序，把单元格区域的内容保存到变量 ss 中。代码如下：

```
>> ss = GetNamedRangeFromExcel('disc','C:\Documents and Settings\ader\My Documents\Book1.xls')
ss =
    [4]    [5]
    [6]    [7]
```

运行结果表明单元格的内容保存到变量 ss 中。格式为单元格式，如果要变成实数矩阵，可以执行命令：

```
>> ss = cell2mat(ss)
ss =
      4      5
      6      7
```

这时 ss 就变成一个实数矩阵。

思考题

1. 将金牛股份(000937)2005 年 12 月 14 日至 2006 年 1 月 10 日的成交记录写入 Excel 中，画出股价图，编写 VBA 程序计算均值、方差及夏普比率。

2. 将金牛股份(000937)2005 年 12 月 14 日至 2006 年 1 月 10 日的成交记录写入 Excel 中，用 VBA 程序编写波段最大跌幅，并将结果和 MATLAB 比较。



第 9 章　MATLAB 操作数据库

金融时间序列数据一般保存在金融数据库中，银行、保险及证券公司等金融机构都有记录交易和客户资料数据库，读者应了解数据库的基本知识，熟悉 MATLAB 操作数据库的常用方法，学会用 SQL 语言对数据库文件进行读取。

9.1　数据库基本原理

9.1.1　数据库工具包

Database Toolbox 提供了同任何支持 ODBC/JDBC 标准的数据库进行数据交换的能力。利用在工具箱中集成的 Visual Query Builder 工具，读者无需学习任何 SQL 语句就可以实现在数据库中查寻数据的功能。这样 MATLAB 就能够对存储在数据库中的数据进行各种各样的复杂分析。在 MATLAB 环境中，也可以使用 SQL 命令来进行数据库操作。

MATLAB 支持下面类型的数据库：

- IBM DB2
- IBM Informix
- Ingres
- Microsoft Access
- Microsoft Excel
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- Sybase SQL Server
- Sybase SQL Anywhere

MATLAB 数据库操作具有如下特点：

- 支持 ODBC/JDBC 数据库连接。
- 支持 SQL 语句。可以在 MATLAB 环境下直接执行 SQL 查询命令。
- 动态数据调入。可以根据需要使用 SQL 获取 MATLAB 中的数据，利用工具箱对某一种类型的数据库进行查询。

- 在 MATLAB 中对数据的调入或调出操作都能保持原有的数据类型。
- 多个对话能力。通过工具箱可在 MATLAB 中从一个数据库中调入数据，对数据进行分析，然后输出到另一个数据库中。
- 处理大量数据的能力。采用本工具箱可以一次或分几次处理大量的数据，这样能高效地进行数据处理。
- 连续状态的数据库连接。一旦和某个数据库的连接建立起来后，数据库一直是打开的，除非在 MATLAB 中执行关闭语句。这提高了数据库的读取速度，减少了不必要的命令来调入、输出数据。
- 无需了解 SQL 也能够对数据库数据进行操作。

MATLAB 提供了强大的连接数据库功能，主要方式有 2 种：

- VQB(Visual Query Builder)；
- Using fuctions。

VQB 为界面菜单操作，只需根据菜单的提示做出相应的选择。在 Command 窗口中输入 querybuilder 命令，即可出现 Visual Query Builder 界面，界面内容比较简单。

Using fuctions 是命令语言，通过输入数据库命令完成对数据库的一系列操作。

9.1.2　ODBC 数据库

ODBC(Open Database Connectivity，开放数据库连接)是 Microsoft 公司为应用程序操作数据库而开发的一种策略，ODBC 的优点是可以使得应用程序按照固定方式操作不同数据库，如 MS Query、Access、Word、Excel、Foxpro、VB、C 等。MATLAB 也可以通过 ODBC 访问数据库。

ODBC 数据源是由数据源名称 DSN(Data Source Name)来引用的，位于支持开放式数据库连接(ODBC)协议的程序或数据库中，它包括 Microsoft SQL Server(但不仅限于此)。例如，SQL Server 的数据源是 SQL Server 数据库以及访问该服务器的网络信息。

ODBC 使用驱动程序连接数据库，不同的数据库需要不同的驱动程序。ODBC 有一个驱动程序管理器，包含在 ODBC.DLL 中，可连接到所有的应用程序中，负责绑定 ODBC 函数与 DLL 函数。

9.1.3　关系型数据库

现在的数据库大多为关系型数据库，关系型数据库最早由 IBM 公司的 E. F. Codd 提出，其特点如下：

- 面向集合，可以同时操作多个行。
- 数据逻辑独立。
- 数据自动导航，访问路径由数据库优化器决定。

- 操作方便。

关系型数据库包含下面3个部分：

- 关系模型的数据结构。关系模型的数据结构为单一的数据结构，基本元素是由行与列构成的二维数表。
- 关系型数据的完整性。关系型数据完整性包括实体完整性、参照完整性和用户定义的完整性。实体完整性是指用主键来唯一标志表中的行与列，主键的任一属性不能为空；参照完整性是指外键或者为空，或者等于它所参照的表的主键的某个值；用户定义完整性是指对某一具体数据库的约束条件。
- 关系型的数据操作。关系型数据库对数据操作能力非常强，函数多。

9.2 VQB 方法连接数据库

9.2.1 Access 数据库介绍

Access 是 Office 办公套件中一个重要组成部分。刚开始时 Microsoft 公司是将 Access 单独作为一个产品进行销售，后来 Microsoft 公司发现如果将 Access 捆绑在 Office 中一起发售，将带来更加可观的利润，于是将 Access 捆绑到 Office 97 中，使其成为 Office 办公套件中的一个重要成员。现在它已经成为 Office 办公套件中不可缺少的部件。自从 1992 年开始销售以来，Access 已经卖出了超过 6 000 万份，现在已经成为世界上流行的桌面数据库管理系统。

后来 Microsoft 公司通过大量的改进，将 Access 新版本功能变得更加强大，不管是处理公司客户订单数据，管理自己个人通讯录，还是记录和处理大量金融数据，人们都可以利用它来解决数据管理与分析工作。

Access 功能这么强，使用起来会不会很麻烦呢？随着版本升级，Access 的使用也变得越来越容易，过去很烦琐的工作现在只需几个很简单步骤就可以解决。

9.2.2 定义 ODBC 数据库

1. 建立 Access 文件

下面以一个例子来说明如何将 Access 中的数据导入到 MATLAB 中，其他数据库如 SQL、Dbase、Foxpro 等也可以类似地和 MATLAB 进行数据连接。在“C:\新建文件夹”先建立了一个 Access 文件，文件名为 db1.mdb，如图 9.1 所示。

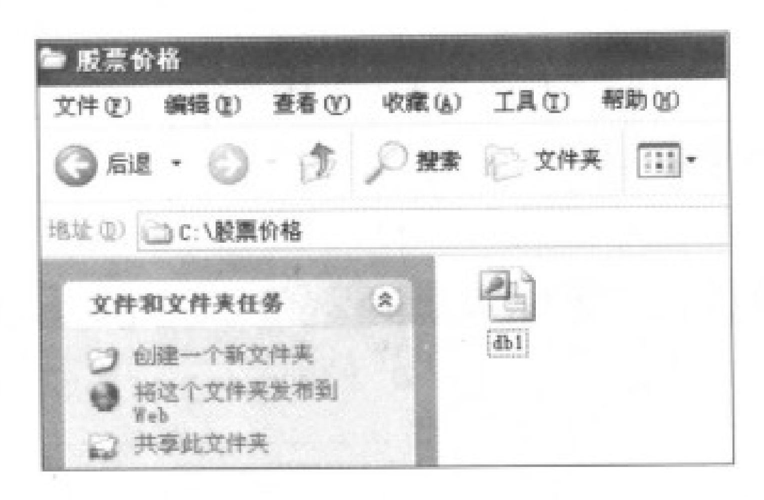


图 9.1 Access 中文件

下面浏览其内容，如图 9.2 所示。

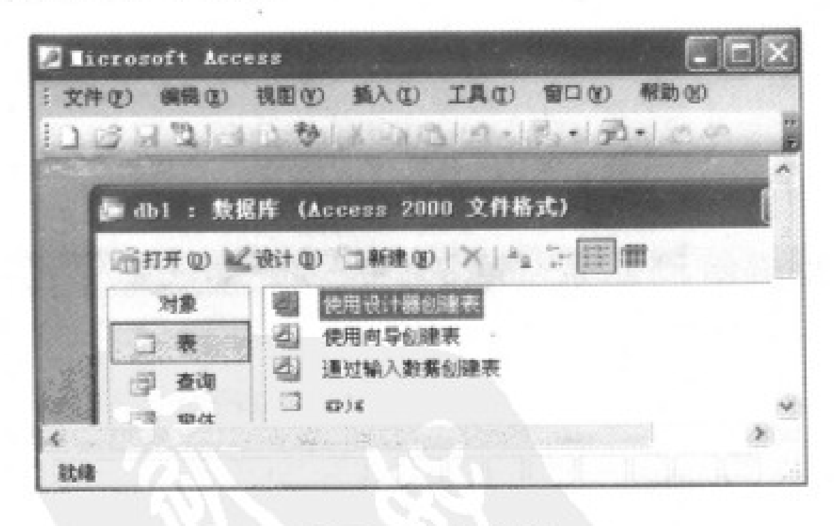
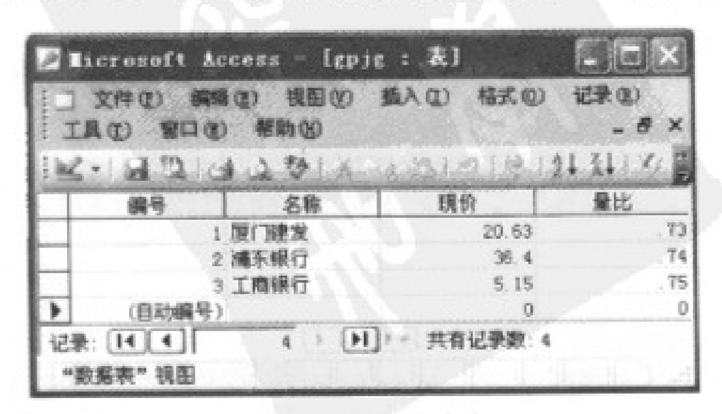


图 9.2 浏览工作簿

在图 9.2 的窗口中单击“□ gpjg”图标，数据库内容如图 9.3 所示。



编号	名称	现价	量比
1	厦门建发	20.63	.73
2	浦东银行	36.4	.74
3	工商银行	5.15	.75
(自动编号)		0	0

图 9.3 数据库内容

文件中只有一个工作簿 gpjg，有 3 条记录，字段名分别为编号、名称、现价和量比。退出 Access。下面利用 MATLAB 自带的 VQB 定义当前数据库。

2. 定义 Access 数据文件

打开 MATLAB，在 Command 窗口下输入命令 querybuilder，如图 9.4 所示。

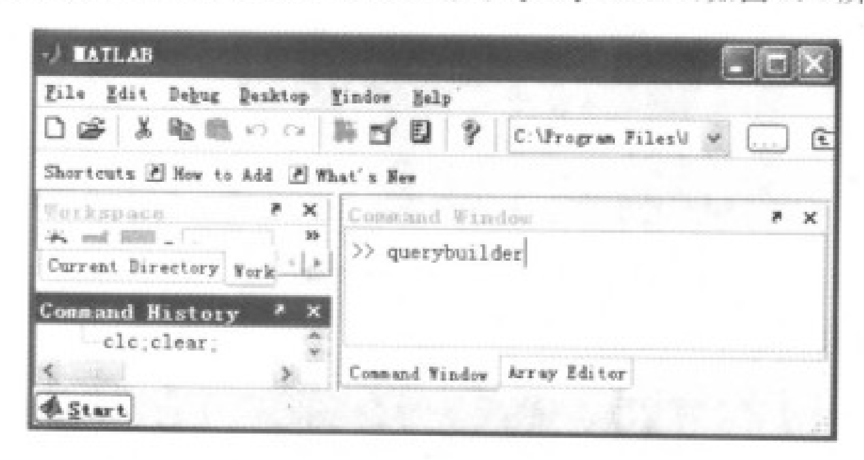


图 9.4　在 MATLAB 环境中调用 VQB

弹出如图 9.5 所示 VQB 对话框。

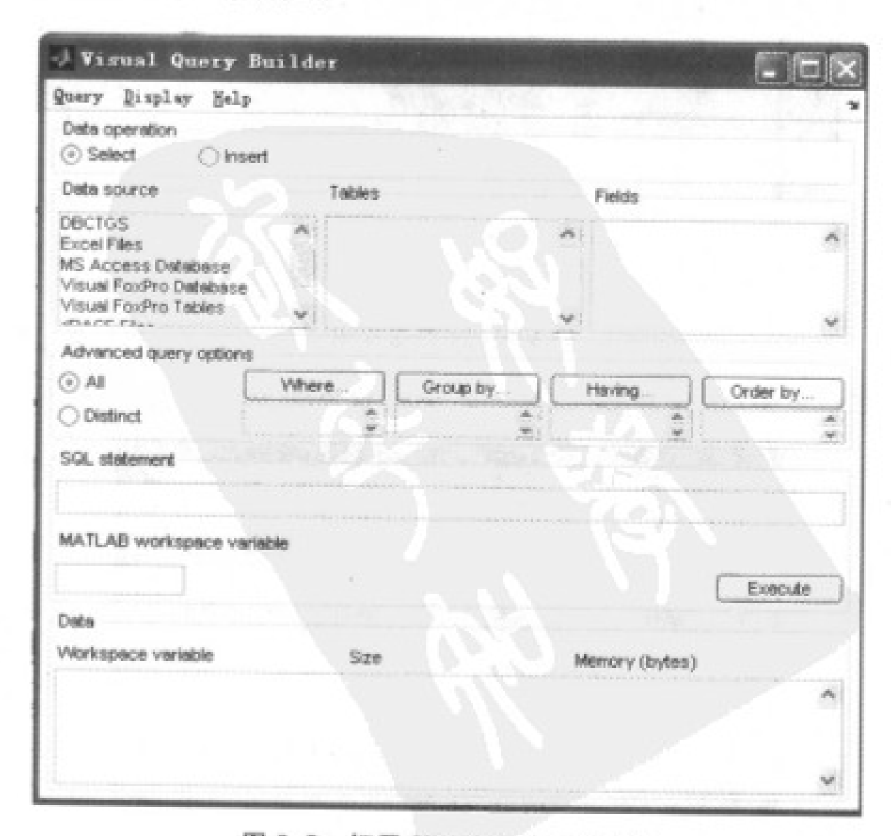


图 9.5　打开 Visual Query Builder

打开 Query 菜单，如图 9.6 所示。

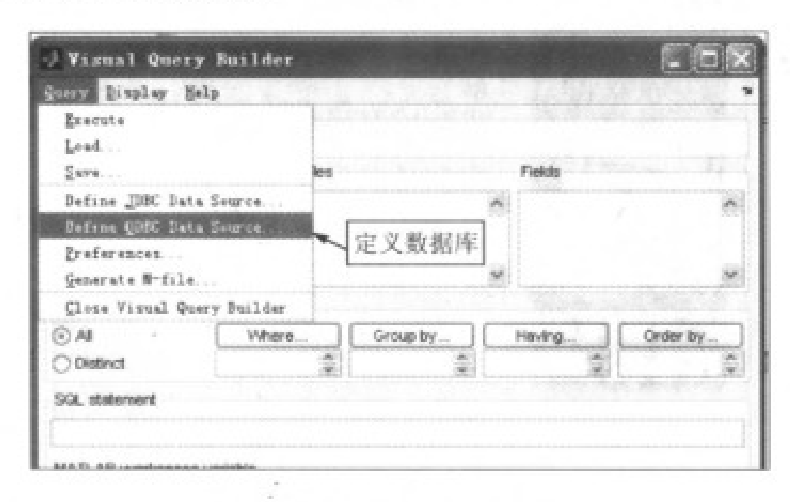


图 9.6 选中 ODBC 数据管理器

“ODBC 数据源管理器”是标准的远程数据库。下面通过 ODBC 数据(Open DataBase Connectivity，开放式数据库连接)访问接口来进行数据操作。首先需要定义 db1.mdb 为当前数据库。在图 9.6 中选择【Define ODBC Data Sourse】菜单项，弹出【ODBC 数据源管理器】，如图 9.7 所示。

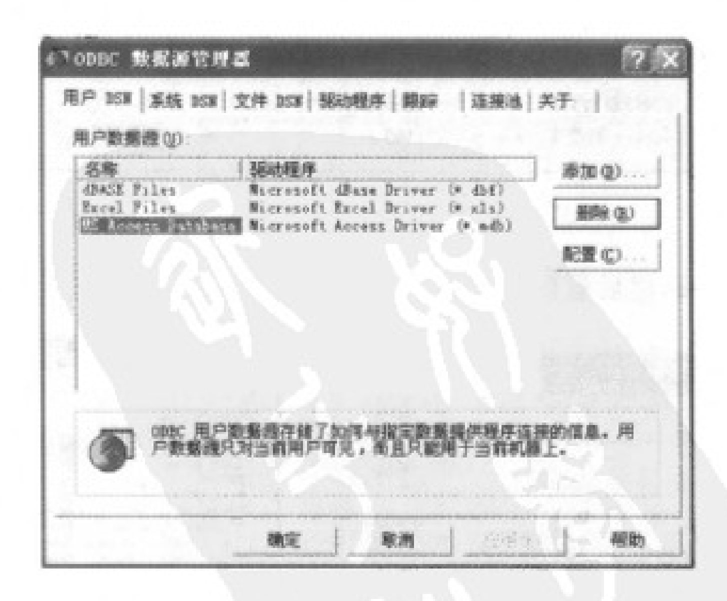


图 9.7 选中 Access 数据库驱动程序

单击【添加】按钮，出现如图 9.8 所示界面。

单击【完成】按钮，在随后出现的页面中单击【选择】按钮，选中“C:\股票价格\db1.mdb”文件，如图 9.9 所示。

单击【确定】按钮返回，在图 9.10 界面中输入文件 db1.mdb 中 gpjg 工作簿。



图 9.8 选中数据安装的驱动程序

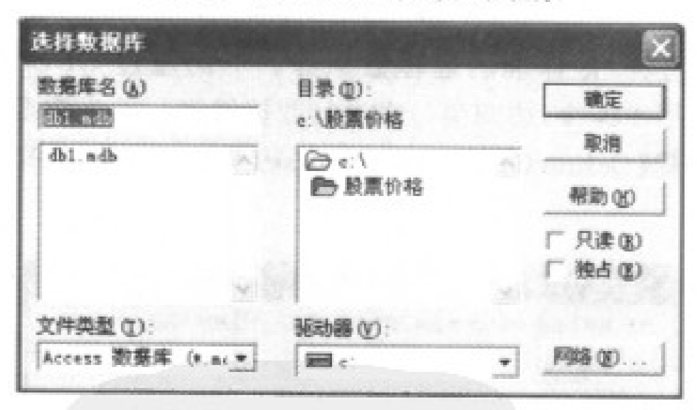


图 9.9 浏览目标文件

单击【确定】按钮返回，然后看到ODBC数据管理器中定义了gpjg工作簿，如图9.11所示。





图 9.10 浏览目标文件中工作簿

图 9.11 添加工作簿图

9.2.3 MATLAB 与 Access 进行数据交换

首先介绍把 Access 数据输入到 MATLAB 中的方法。

在图 9.11 窗口中，单击【确定】按钮后退出，出现图 9.12 的界面。

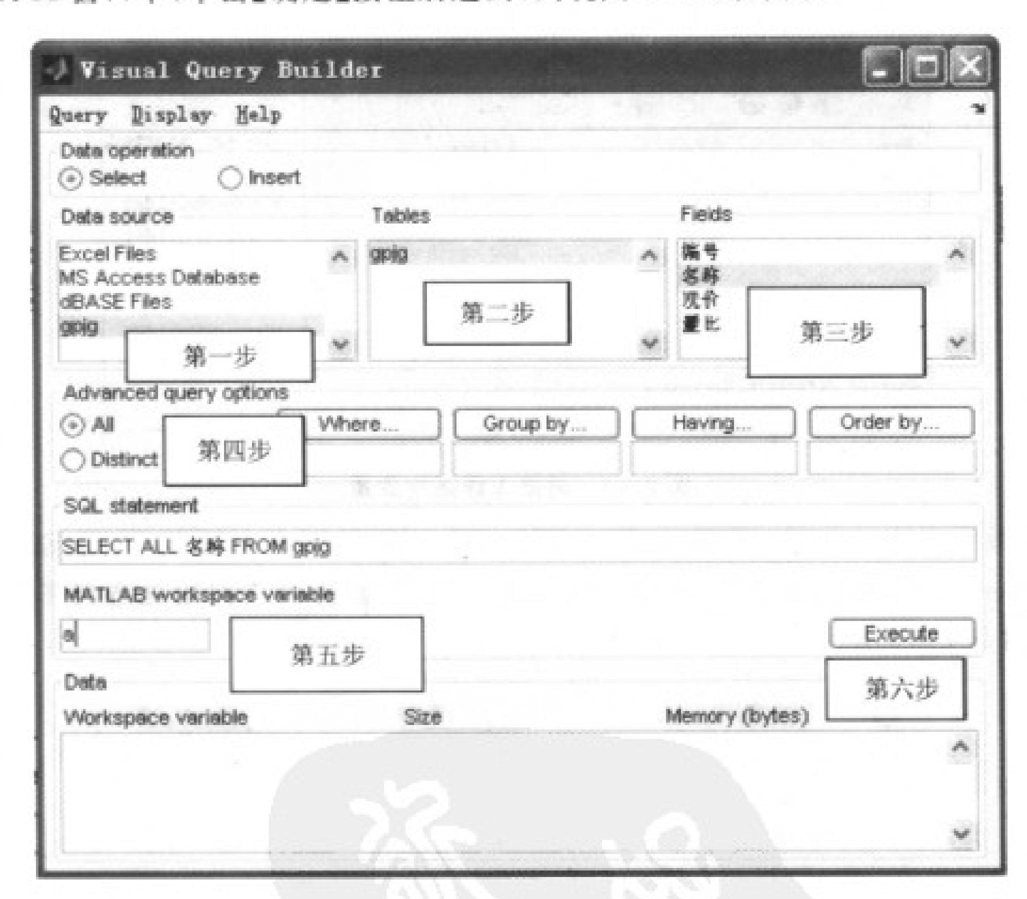


图 9.12 将工作簿中字段数据导入 MATLAB 工作区中

按照图 9.12 的次序分别单击相关按钮。下面介绍每一步的内容。

第一步是选中“gpjg”工作簿作为数据来源。

第二步把“Tables”中选中 gpjg 作为当前工作簿。

第三步中“Fields”是字段名，选择“名称”。

第四步对字段名下数据进行选择，这里选择 All，将“名称”字段下数据全部选中。也可以根据右边的几个菜单进行选择。

第五步填入在 MATLAB 中变量“a”，用以保存上面选中的数据。

第六步单击【Excute】按钮，这样字段“名称”下数据全部进入 MATLAB 的“a”变量中。

注意【SQL statement】一栏下“SELECT ALL 名称 FROM gpjg”是根据操作自动生成的 SQL 语句，不用理会。

浏览 MATLAB 工作区可以看到变量 a，注意 a 是单元数据，如图 9.13 所示。

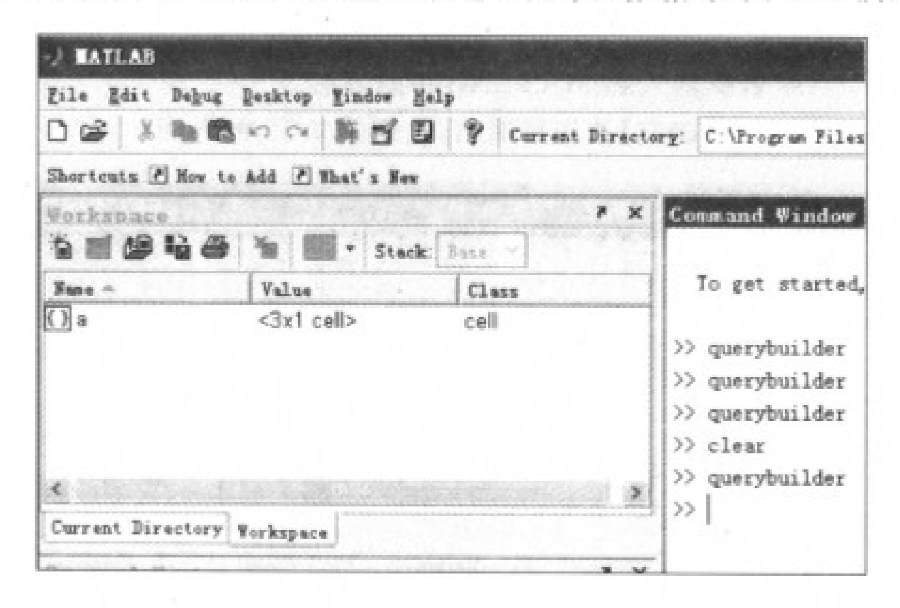


图 9.13 浏览工作区中变量

有时在数据库读入和写入数据出现空记录 null，这时就要事先设定处理方法。在 Visual Query Builder 窗口的【Query】菜单中选择【Preference】，就会自动弹出处理异常数据的对话框，具体如图 9.14 所示。

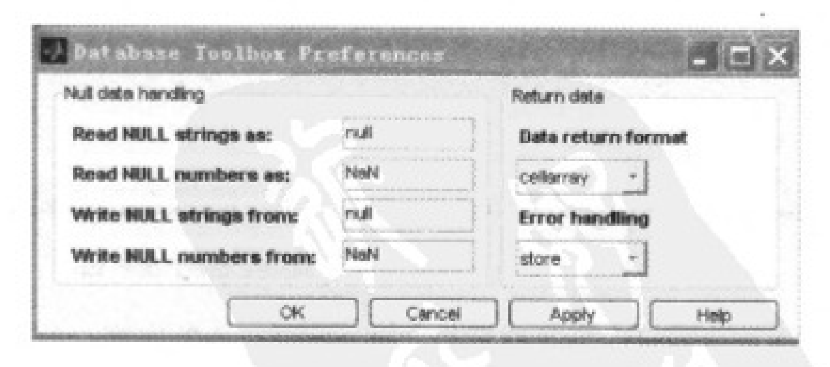


图 9.14 处理异常数据对话框

【Data return format】菜单栏提供了可以选择的输出变量的类型。

- cellarry　　单元型数组
- numeic　　数字型
- structure　　结构变量

【Error handing】菜单栏提供了出现错误时的处理信息。

- store　　保存错误信息
- report　　报告错误信息
- empty　　不处理

如果需要把MATLAB中的变量内容导入Access文件，例如MATLAB Workspace中的close变量保存了收盘价，查看收盘价，则执行如下命令：

```
>> new = {'东方航空',4.35,0.67}
new =
    '东方航空'    [4.3500]    [0.6700]
```

下面在数据库中添加该记录。回到图9.12窗口中，选中【insert】单选按钮，具体如图9.15所示。

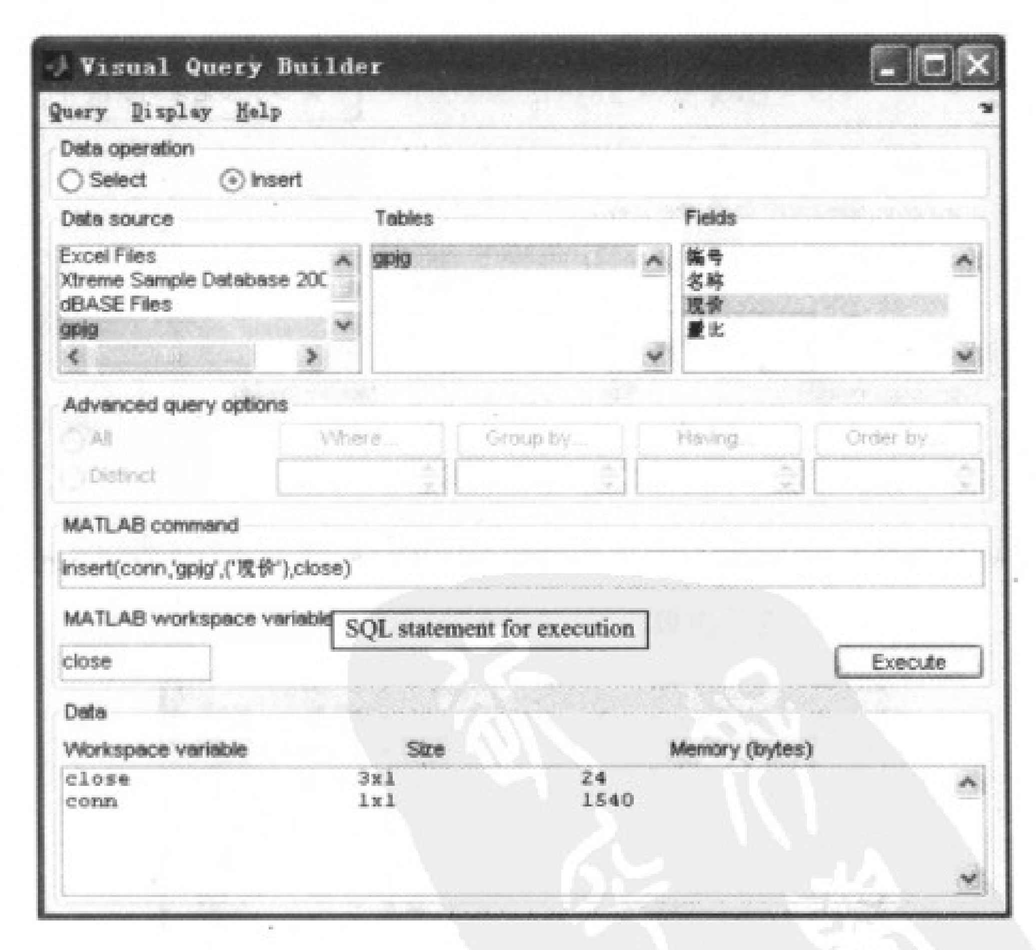


图9.15　更改数据库记录操作界面

在图9.15窗口用鼠标在Fields栏下选择添加的字段，在MATLAB workspace variable栏下输入变量new。

图9.16中SQL语句是机器自动添加的，单击【Execute】按钮，打开表gpjg可以看到东方航空的数据被成功添加进数据库。新增内容如图9.17所示。



图 9.16 新增记录的步骤

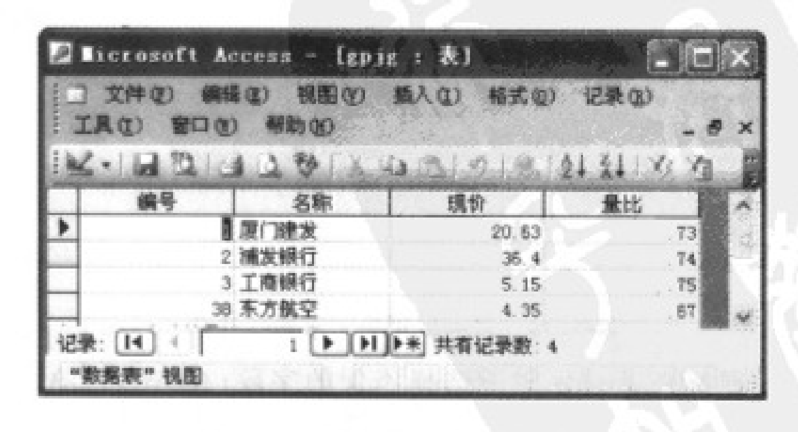


图 9.17 新增记录的数据表内容

9.3 利用SQL语句访问数据库

SQL(Structured Query Language)是一种操作数据库的结构化语言，由于其功能强大，使用方便，1986年10月美国国家标准学会(ANSI)批准将SQL作为美国数据库的标准语言，随后美国国际标准化组织(ISO)也做出同样的决定。

SQL提供了很多语句来查询数据库，但是核心的语句有4种，包含了对数据库增删查改4项基本内容。

- Select 在数据库中查询。
- Delete 用于删除数据库中的记录。
- Insert 用于向数据库添加新记录。
- Update 用于更新数据库中的新记录。

9.3.1 数据库连接

1. 查询当前注册的数据库名称

在MATLAB中查询当前获得注册数据库列表的命令是getdatasources。

```
>> datasources = getdatasources
datasources =
           'gpjg'
```

说明有一个名称为gpjg注册数据库。

2. 建立和数据库的连接

调用方式

```
conn = database('datasourcename', 'username', 'password')
conn = database('databasename', 'username', 'password', 'driver','databaseurl')
```

输入参数

datasourcename	当前定义的数据库
username	用户名
password	密码
driver	驱动器

如果没用用户名和密码，则用''代替。例如建立和gpjg数据库的连接，可以输入下面命令：

```
>> conn = database('gpjg','','')                %没有用户名与密码
conn =
        Instance: 'gpjg'
        UserName: ''
          Driver: []
             URL: []
     Constructor: [1x1 com.mathworks.toolbox.database.databaseConnect]
         Message: []
          Handle: [1x1 sun.jdbc.odbc.JdbcOdbcConnection]
         TimeOut: 0
      AutoCommit: 'on'
            Type: 'Database Object'
```

调用方式：

```
conn = database('datasourcename', 'username', 'password')
conn = database('databasename', 'username', 'password',
'driver','databaseurl')
```

可以用 isconnection 命令查看连接的状况。

```
>> isconnection(conn)
ans =
      1
```

ans=1 说明 MATLAB 处于和数据库连接的状态。
进一步查看数据库是否是只读的。

```
>> isreadonly(conn)
ans =
      1
```

ans=1 说明是链接的数据库是只读的。
如果需要更多的信息，则可以用 dmd 命令查看连接。

```
>> dbmeta = dmd(conn)
dbmeta =
        DMDHandle: [1x1 sun.jdbc.odbc.JdbcOdbcDatabaseMetaData]
```

下面用 get 查看 dbmeta 的内容。

```
>> v = get(dbmeta)
v =
                              AllProceduresAreCallable: 1
```

```
          AllTablesAreSelectable: 1
DataDefinitionCausesTransactionCommit: 1
  DataDefinitionIgnoredInTransactions: 0
          DoesMaxRowSizeIncludeBlobs: 0
                 Catalogs: {'C:\股票价格\db1'}
                    CatalogSeparator: '.'
                    CatalogTerm: '数据库'
              DatabaseProductName: 'ACCESS'
          DatabaseProductVersion: '04.00.0000'
                               ......
                UsesLocalFilePerTable: 0
                       UsesLocalFiles: 1
```

注意v是结构变量,例如Catalogs特征可以显示数据库的来源,可以执行如下操作:

```
>> v.Catalogs
ans =
    'C:\股票价格\db1'
```

用support命令了解可以对数据库进行哪些操作。

```
>> a = supports(dbmeta);
```

利用ping函数可以查看conn的情况,代码如下:

```
>> ping(conn)
ans =
      DatabaseProductName: 'ACCESS'
   DatabaseProductVersion: '04.00.0000'
            JDBCDriverName: 'JDBC - ODBC Bridge (odbcjt32.dll)'
         JDBCDriverVersion: '2.0001 (04.00.6304)'
   MaxDatabaseConnections: 64
           CurrentUserName: 'admin'
               DatabaseURL: 'jdbc:odbc:gpjg'
   AutoCommitTransactions: 'True'
```

用close函数可以关闭同数据库的连接。代码如下:

```
>> close(conn)
```

9.3.2 MATLAB数据库操作简介

MATLAB的数据库操作是面向对象的工具箱,含有下面的几种类型的对象:

- Cursor
- Database
- Database metadata
- Driver
- Drivermanager
- Resultset
- Resultset metadata

每个对象都有自己的目录，位于 toolbox/database/database 中，子目录前带有@符号。

下面浏览在“C:\股票价格”目录下的 Access 数据库文件 db1.mdb，数据库中含有 price 变量，内容如图 9.18 所示。

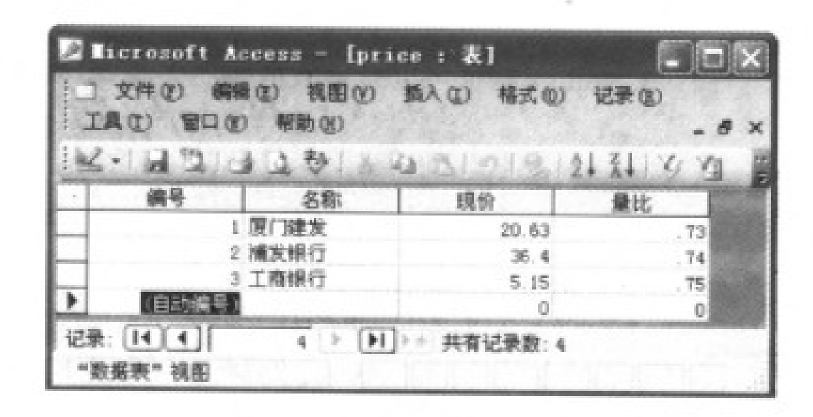


图 9.18 数据库文件内容

在操作数据库之前必须对数据库文件进行注册，现将其命名为 gpjg。

9.3.3 在 MATLAB 中使用 SQL 语句操作数据库

1. 读取数据表内容

在 MATLAB 中的 exec 可以执行 SQL 数据库操作命令。下面是一个例子。

```
>> curs = exec(conn,'sqlqurey')
curs =
        Attributes: []
              Data: 0
    DatabaseObject: [1x1 database]
          RowLimit: 0
          SQLQuery: 'sqlqurey'
           Message: [1x105 char]
```

```
          Type: 'Database Cursor Object'
     ResultSet: 0
        Cursor: 0
     Statement: 0
         Fetch: 0
```

上述语句的内容是执行SQL中的sqlquery命令,返回值curs是一个对象,包含了数据库的众多信息。

【例9-1】 读取数据库中price数表中的所有内容。

```
>> str = 'select * from price';                %SQL输出数据库所有内容命令
>> curs = exec(conn,str)
curs =
        Attributes: []
              Data: 0
    DatabaseObject: [1x1 database]
          RowLimit: 0
          SQLQuery: 'select * from price'
           Message: []
              Type: 'Database Cursor Object'
         ResultSet: [1x1 sun.jdbc.odbc.JdbcOdbcResultSet]
            Cursor: [1x1 com.mathworks.toolbox.database.sqlExec]
         Statement: [1x1 sun.jdbc.odbc.JdbcOdbcStatement]
             Fetch: 0
```

下面用fetch函数查看其内容。

```
>> x = fetch(curs)
x =
        Attributes: []
              Data: {3x4 cell}
    DatabaseObject: [1x1 database]
          RowLimit: 0
          SQLQuery: 'select * from price'
           Message: []
              Type: 'Database Cursor Object'
         ResultSet: [1x1 sun.jdbc.odbc.JdbcOdbcResultSet]
            Cursor: [1x1 com.mathworks.toolbox.database.sqlExec]
         Statement: [1x1 sun.jdbc.odbc.JdbcOdbcStatement]
```

```
            Fetch: [1x1 com.mathworks.toolbox.database.fetchTheData]
```

列出 x.Data 的内容。

```
>> x.Data
ans =
    [1]    '厦门建发'    [20.6300]    [0.7300]
    [2]    '浦发银行'    [36.4000]    [0.7400]
    [3]    '工商银行'    [ 5.1500]    [0.7500]
```

也可以用 get 函数获取数据。

```
> get(x,'Data')
ans =
    [1]    '厦门建发'    [20.6300]    [0.7300]
    [2]    '浦发银行'    [36.4000]    [0.7400]
    [3]    '工商银行'    [ 5.1500]    [0.7500]
```

【例 9-2】 获取数据库字段名下的记录。

```
>> setdbprefs('DataReturnFormat','cellarray')
>> curs = exec(conn, 'select  名称 from price')
curs =
        Attributes: []
              Data: 0
    DatabaseObject: [1x1 database]
          RowLimit: 0
          SQLQuery: 'select  名称 from price'
           Message: []
              Type: 'Database Cursor Object'
         ResultSet: [1x1 sun.jdbc.odbc.JdbcOdbcResultSet]
            Cursor: [1x1 com.mathworks.toolbox.database.sqlExec]
         Statement: [1x1 sun.jdbc.odbc.JdbcOdbcStatement]
             Fetch: 0
```

获得 curs 中的数据。

```
>> curs = fetch(curs, 2)
curs =
        Attributes: []
              Data: {2x1 cell}
    DatabaseObject: [1x1 database]
```

```
        RowLimit: 0
        SQLQuery: 'select  名称 from price'
         Message: []
            Type: 'Database Cursor Object'
       ResultSet: [1x1 sun.jdbc.odbc.JdbcOdbcResultSet]
          Cursor: [1x1 com.mathworks.toolbox.database.sqlExec]
       Statement: [1x1 sun.jdbc.odbc.JdbcOdbcStatement]
           Fetch: [1x1 com.mathworks.toolbox.database.fetchTheData]
>> c = curs.Data
c =
    '厦门建发'
    '浦发银行'
```

可以用columnnames函数列出字段名。

```
>> columnnames(curs)
ans =
     '名称
```

2. 更新数据库记录

用MATLAB单元数组数据更新数据库表的数据可以使用update函数。

调用方式

```
update(conn, '表名', 字段名, exdata, 'where 语句')
update(conn, '表名', 字段名, {datA,datAA,…; datB,datBB,…; datn,datnn}, {'where 语句'})
```

例如将表price中现价小于6元的股票现价改为7元。

```
>> update(conn,'price',{'现价'},{7},'where 现价<6')
```

3. 添加新记录

1) fastinsert函数

调用方式

```
fastinsert(conn, 'tablename', colnames, exdata)
```

输入参数

'tablename'	数表的名称
colnames	字段名

exdata　　　　MATLAB 中的变量

下面添加一个新记录。

```
>> new = {'东方航空',4.35,0.67}
new =
        '东方航空'    [4.3500]    [0.6700]
>> fastinsert(conn,'price',{'名称','现价','量比'},new)
>> commit(conn)                % 确认操作
```

图 9.19 所示为更改后的数据文件。

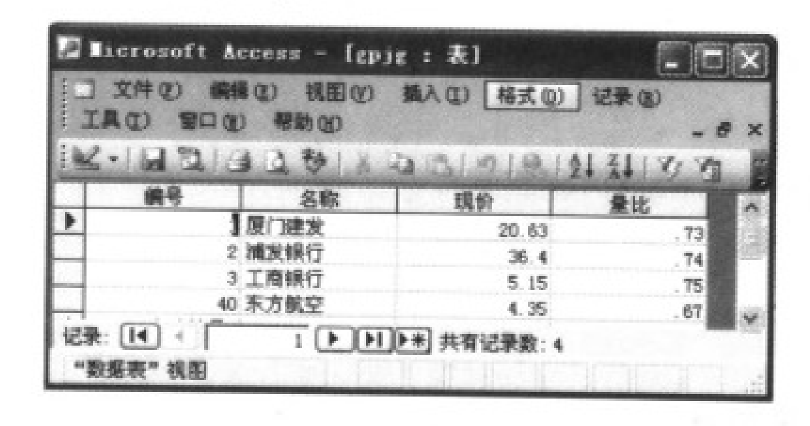


图 9.19　更改后的数据文件

2) insert 函数

调用方式

```
insert(conn, 'tab', colnames, exdata)
```

输入参数同前。

4. 撤销对数据库的操作

有时需要撤销新近的数据库操作，这时可以利用 rollback 函数。

调用方式

```
rollback(opperation)
```

输入参数

operation　　　　新近的操作命令

例如将表 price 中现价小于 6 元的股票现价改为 7 元。

```
>> update(conn,'price',{'现价'},{7},'where 现价<6')
```

如果需要撤销上述操作，则可以执行下面操作。

```
>> rollback(conn)
```

5. 得到自动更新表列

调用方式 1

```
vl = versioncolumns(dbmeta, 'cata')
```

返回所有更新的列数据。

```
vl = versioncolumns(dbmeta, 'cata', 'sch')
vl = versioncolumns(dbmeta, 'cata', 'sch', 'tab')
```

输入参数

dbmeta	数据库对象
cata	数据库类型
sch	数据库名称
tab	表的名称
sal	变量名

输出参数

```
    vl
>> conn = database('gpjg','','')
>> dbmeta = dmd(conn)    % 构造数据库元对象
>> v = get(dbmeta);
>> vl = versioncolumns(dbmeta, '名称')
vl =
      {}
```

输出内容为空时说明列数据没有改变。

6. 获取数据库的字段名

调用方式

```
A = columns(dbmeta, 'cata')
A = columns(dbmeta, 'cata', 'sch')
A = columns(dbmeta, 'cata', 'sch', 'tab')
```

输入参数

dbmeta 数据库元对象
'cata' 类
'sch' 结构
'tab' 数据库中表的名称

输出参数

A 数据表的字段名

方法 1

```
>> z = columns(dbmeta,[],[],'price')
z =
    '编号'    '名称'    '现价'    '量比'
```

方法 2

```
>> fie = columns(dbmeta,[],[])
fie =
    'MSysAccessObjects'    {1x2 cell}
    'MSysAccessXML'        {1x6 cell}
    'price'                {1x4 cell}
```

fie 是一个 3 行 2 列的单元数组，每行分别是表的名称与其含有的字段名。

```
>> ss = fie(3,2);
>> ss{1,:}
ans =
    '编号'    '名称'    '现价'    '量比'
```

7. 获取数据库的数据的程序

【例 9-3】 下面是一个读取 Access 数据库的小文件。代码如下：

```
function dbimportdemo()
% 确定和数据库连接的最大时间。
timeoutA = logintimeout(5)
% 建立和数据库的连接
% 打开数据源
connA = database('gpjg','','')
ping(connA)
% 执行 SQL 查询语句
cursorA = exec(connA,'select * from price');
% 获取前 2 行数据
```

```
cursorA = fetch(cursorA,2)
% 显示前 2 行内容
AA = cursorA.Data
% 关闭连接
close(cursorA)
close(connA)
```

思考题

1. 将金牛股份(000937)2005 年 12 月 14 日至 2006 年 1 月 10 日的交易记录保存到 Excel 中，编写程序将数据读入 MATLAB 中，进一步将数据读入 Access 数据库文件。

2. 讨论如何将金牛股份 2001 年至 2006 年的季报、年报数据保存到 Access 数据库中，考虑如何构建数据库的架构。



附　录

附录 1

《财政部、中国人民银行、中国证券监督管理委员会关于试行国债净价交易有关事宜的通知》(财库[2001]12 号)

全国银行间同业拆借中心,上海证券交易所,深圳证券交易所,中央国债登记结算有限责任公司:

为促进国债二级市场发展,实现国债交易方式与国际通行做法的逐步接轨,现决定在全国银行间债券市场、上海证券交易所、深圳证券交易所实行国债净价交易。现将有关事宜通知如下:

一、净价交易概念

净价交易是指在现券买卖时,以不含有自然增长应计利息的价格报价并成交的交易方式。在净价交易条件下,由于国债交易价格不含有应计利息,其价格形成及变动能够更加准确地体现国债的内在价值、供求关系及市场利率的变动趋势。

二、实行净价交易应具备的技术条件

实行净价交易对国债交易报价系统及交易清算系统提出了新的要求,主要内容是:报价系统同时显示国债全价、净价及应计利息额;交易清算及交割单打印系统自动计算应计利息额并在交割单上分别列明结算价、净价及应计利息额(净价+应计利息额=结算价)。

三、应计利息额计算方法

应计利息额=票面利率÷365 天×已计息天数

上述公式各要素具有以下含义:

1. 应计利息额。零息国债是指发行起息日至交割日所含利息金额;附息国债是指本付息期起息日至交割日所含利息金额。

2. 票面利率。固定利率国债是指发行票面利率;浮动利率国债是指本付息期计息利率。

3. 年度天数及已计息天数。一年按 365 天计算,闰年 2 月 29 日不计算利息(下同);已计息天数是指起息日至交割当日实际日历天数。

4. 当票面利率不能被 365 天整除时,计算机系统按默认位数全额计算,交割单应计利息总额按“四舍五人”原则,以元为单位保留 2 位小数列示。

5. 国债交易以每百元国债价格进行报价,应计利息额也须按每百元国债所含利息额列示。

四、具体实施办法及步骤

国债净价交易是新的交易方式,对证券市场和相关领域会产生较大影响,也对计算机技术

有较高的要求。试行国债净价交易应遵循积极稳妥的原则分步实施。具体为:第一步继续按现行方式实行全额报价交易,但报价系统应同时实行全价价格、净价价格和应计利息额三项要素中至少两项,并对交割单打印系统进行修改,交割单上须分别列明结算价、净价及应计利息额;第二步,试行净价报价,以成交价格和应计利息额之和作为结算价格。

为尽快试行净价交易,各有关单位应要求加紧设计软件和调试计算机系统,基本达到第一步要求的单位可先试行全额报价下的净价交易,条件具备的单位也可直接进入第二步。在2001年12月31日前,各有关单位均要达到净价交易的第二步要求,实现完全的净价交易。

特此通知。

二〇〇一年一月十七日　(中央法规)

附录2

《中国人民银行关于完善全国银行间债券市场债券到期收益率计算标准有关事项的通知》(银发[2007]200号)

全国银行间同业拆借中心、中央国债登记结算有限责任公司:

经研究,现决定对全国银行间债券市场的到期收益率计算标准进行调整,具体如下:

一、全国银行间债券市场到期收益率的日计数基准由“实际天数/365”调整为“实际天数/实际天数”,即应计利息天数按当期的实际天数计算(算头不算尾),闰年2月29日计算利息,付息区间天数按实际天数计算(算头不算尾)。

二、对《中国人民银行关于全国银行间债券市场债券到期收益率计算标准有关事项的通知》(银发[2004]116号)第三条“债券到期收益率计算”中的有关计算公式进行相应调整,详见附件。

调整后的全国银行间债券市场到期收益率计算标准适用于全国银行间债券市场的发行、托管、交易、结算、兑付等业务。

本通知未尽事宜依照《中国人民银行关于全国银行间债券市场债券到期收益率计算标准有关事项的通知》(银发[2004]116号)的有关规定执行。

请你们将本通知及调整后的计算标准通过中国货币网和中国债券信息网向市场成员公告。同时,你们要抓紧做好相应的技术准备工作,最晚于2007年12月1日前开始按照调整后的到期收益率计算标准计算全国银行间债券市场所有产品的到期收益率。

附件:《全国银行间债券市场债券到期收益率计算标准调整对照表》

中国人民银行

二〇〇七年六月二十日

附录3

《上交所关于实施“试行国债净价交易”有关事宜的通知》

各会员单位：

根据财政部、中国人民银行和中国证券监督管理委员会《关于试行国债净价交易有关事宜的通知》(财库[2001]12号)精神，现将有关事宜通知如下：

一、实施“试行国债净价交易”时间从2002年3月25日(星期一)起实施。

二、净价交易概念净价交易是指在现券买卖时，以不含有自然增长应计利息的价格报价并成交的交易方式。在净价交易条件下，由于国债交易价格不含有应计利息，其价格形成及变动能够更加准确地体现国债的内在价值、供求关系及市场利率的变动趋势。

三、实行国债净价交易后，应计利息额的计算方法如下：

应计利息额＝票面利率÷365(天)×已计息天数

上述公式各要素具有以下含义：

1. 应计利息额：零息国债是指发行“起息日”至“成交日”所含利息金额；附息国债是指本付息期“起息日”至“成交日”所含利息金额。

2. 票面利率：固定利率国债是指发行票面利率；浮动利率国债是指本付息期计息利率。

3. 年度天数及已计息天数：1年按365天计算，闰年2月29日不计算利息；已计息天数是指“起息日”至“成交日”实际日历天数。

4. 当票面利率不能被365天整除时，计算机系统按每百元利息额的精度(小数点后保留8位即数据类型位为N15.8)计算；交割单所列“应计利息额”按“四舍五入”原则，以元为单位保留2位小数列示。

5. 国债交易计息原则是“算头不算尾”，即“起息日”当天计算利息，“到期日”当天不计算利息；交易日挂牌显示的“每百元应计利息额”是包括“交易日”当日在内的应计利息额；若国债持有到期，则应计利息额是自“起息日”至“到期日”(不包括到期日当日)的应计利息额。

四、国债净价交易遵循的原则

实行国债净价交易仍将遵守本所现行的交易规则，其次：

1. 实行净价申报和净价撮合成交，以成交价格和应计利息额之和作为结算价格。

2. 国债净价交易以每百元国债价格进行报价，应计利息额按每百元国债所含利息额列示。

3. 报价系统和行情发布系统同时显示净价价格和应计利息额。

4. 在场内申报终端和各证券营业部的行情系统显示的“应计利息额”均为参考值，各会员单位应以中国证券登记结算有限责任公司上海分公司公布的“当日国债品种应计利息额”数据文件的数据为准进行资金清算。

5. 交易清算及交割单打印系统，应自动计算应计利息额并在交割单上分别列明“结算

价”、“净价”及“应计利息额”。

国债净价交易是新的交易方式，对证券市场和相关领域会产生较大影响，因此，各有关单位应遵循积极稳妥的原则，按要求做好各项准备工作，确保试行国债净价交易的顺利实施。

附：《上海证券交易所国债净价交易技术方案》

上海证券交易所

二〇〇二年三月十八日

附录 4

《深交所关于实施国债净价交易有关事宜的通知》

各会员单位：

根据财政部、中国人民银行、中国证券监督管理委员会财库[2001]12 号文通知的精神，现将国债净价交易有关事宜通知如下：

一、实施国债净价交易时间从 2002 年 3 月 25 日（星期一）起实施。

二、净价交易概念

净价交易是指在现券买卖时，以不含有自然增长应计利息的价格报价并成交的交易方式。在净价交易条件下，由于国债交易价格不含有应计利息，其价格形成及变动能够更加准确地体现国债的内在价值、供求关系及市场利率的变动趋势。

三、实行国债净价交易遵循的原则

实行国债净价交易仍遵循本所现行交易规则，以每百元国债价格（净价）进行申报并撮合成交，以成交价格和应计利息额之和作为结算价格；报价系统同时显示净价价格和每百元应计利息额；交易清算及交割单打印系统自动计算应计利息额并在交割单上分别列明结算价、净价及应计利息额。

四、应计利息额的计算方法

应计利息额＝票面利率÷365（天）×已计息天数

上述公式各要素具有以下含义：

1. 应计利息额。零息国债是指发行起息日至成交日所含利息金额；附息国债是指本付息期起息日至成交日所含利息金额。

2. 票面利率。固定利率国债是指发行票面利率；浮动利率国债是指本付息期计息利率。

3. 年度天数及已计息天数。1 年按 365 天计算，闰年 2 月 29 日不计算利息（下同）；已计息天数是指起息日至成交当日实际日历天数。

4. 当票面利率不能被 365 天整除时，计算机系统按每百元利息额的精度（小数点后保留 8 位）计算，交割单应计利息总额按“四舍五入”原则，以元为单位保留 2 位小数列示。

5. 国债交易的计息原则是“算头不算尾”，即起息日当天计算利息，到期日当天不计算利息；交易日挂牌显示的每百元应计利息额是包括交易当日在内的应计利息额。

国债净价交易与现行的国债交易模式有较大差别，请各会员单位务必按要求做好有关准备工作，确保国债净价交易的顺利实施。

附件：《深圳证券交易所国债净价交易技术方案》。

深圳证券交易所
二〇〇二年三月十五日

附录5

《中国银监会关于建立银行业金融机构市场风险管理计量参考基准的通知》
（银监发[2007]48号）

各银监局，各政策性银行、国有商业银行、股份制商业银行、邮储银行：

为进一步加强银行业金融机构市场风险管理，尽快建立银行业金融机构市场风险管理计量参考基准，现就有关要求通知如下：

一、各行应高度重视市场风险管理工作，切实加强对人民币债券收益率曲线（以下简称收益率曲线）的研究和应用。

二、各行可使用自行编制的或其他机构编制的收益率曲线进行市场风险管理，所使用的收益率曲线对市场变化情况的反映要客观、合理。

三、各行用于市场风险管理的收益率曲线应力求建立在全面、客观、合理的债券数据源基础上，并尽量剔除异常价格的影响。

四、各行用于市场风险管理的收益率曲线的构建模型应适应债券市场多变的收益率形态，并满足光滑性和稳定性的要求。

五、各行用于市场风险管理的收益率曲线应至少有一年以上的收益率曲线数据可供比较、查询。

六、各行用于市场风险管理的收益率曲线应包括但不限于国债、央行票据、政策性金融债等人民币债券品种。

七、从2007年10月第一个工作日开始，各行应采用中央国债登记结算有限责任公司编制公布的银行间国债收益率曲线、央行票据收益率曲线与政策性金融债收益率曲线（以下简称中债收益率曲线）计算交易账户人民币头寸市值，并与根据自行编制或其他机构编制的收益率曲线计算得出的相应市值在每个工作日至少进行一次比较。如果在每个季度内有5个（含）以上工作日两者计算结果相差超过1％，则应在下个季度前10个工作日内向中国银监会书面报告，并做出详细、准确的说明。

八、直接使用中债收益率曲线对交易账户人民币头寸进行市值计算的银行，不适用第七条规定。

九、鼓励各行参照第七条规定的中债收益率曲线计算风险价值(VaR)等市场风险管理数据,对所承担的市场风险水平进行量化计算。

十、境内外资法人银行、外国银行分行适用本通知各项规定。城市商业银行、农村商业银行、农村合作银行、城市信用社等其他银行业金融机构可根据具体情况参照执行。

十一、属地监管的银行业金融机构法人在按照第七条规定向银监会报告的同时,应将报告抄送所在地银行业监管机构。

请各银监局将本通知转发至辖内各法人银行业金融机构。

中国银监会

二〇〇七年五月二十日

附录6

雅虎(yahoo)财经网站支持的交易所

国家及地区	交易所名称	股票后缀	行情延迟时间	数据来源
United States of America	American Stock Exchange	不要后缀	20 min	Direct from Exchange
United States of America	Chicago Board of Trade	.CBT	10 min	Comstock
United States of America	Chicago Mercantile Exchange	.CME	10 min	Comstock
United States of America	NASDAQ Stock Exchange	N/A	15 min	Direct from Exchange
United States of America	New York Board of Trade	.NYB	30 min	Comstock
United States of America	New York Commodities Exchange	.CMX	30 min	Comstock
United States of America	New York Mercantile Exchange	.NYM	30 min	Comstock
United States of America	New York Stock Exchange	不要后缀	20 min	Direct from Exchange
United States of America	OTC Bulletin Board Market	.OB	20 min	Direct from Exchange
United States of America	Pink Sheets	.PK	15 min	Direct from Exchange
Argentina	Buenos Aires Stock Exchange	.BA	30 min	Comstock
Austria	Vienna Stock Exchange	.VI	15 min	Telekurs
Australia	Australian Stock Exchange	.AX	20 min	Comstock
Brazil	Sao Paolo Stock Exchange	.SA	15 min	Comstock
Canada	Toronto Stock Exchange	.TO	15 min	Comstock

续表

国家及地区	交易所名称	股票后缀	行情延迟时间	数据来源
Canada	TSX Venture Exchange	.V	15 min	Comstock
China	Shanghai Stock Exchange	.SS	30 min	Direct from Exchange
China	Shenzhen Stock Exchange	.SZ	30 min	Direct from Exchange
Denmark	Copenhagen Stock Exchange	.CO	15 min	Telekurs
France	Paris Stock Exchange	.PA	15 min	Telekurs
Germany	Berlin Stock Exchange	.BE	15 min	Telekurs
Germany	Bremen Stock Exchange	.BM	15 min	Telekurs
Germany	Dusseldorf Stock Exchange	.DU	15 min	Telekurs
Germany	Frankfurt Stock Exchange	.F	15 min	Telekurs
Germany	Hamburg Stock Exchange	.HM	15 min	Telekurs
Germany	Hanover Stock Exchange	.HA	15 min	Telekurs
Germany	Munich Stock Exchange	.MU	15 min	Telekurs
Germany	Stuttgart Stock Exchange	.SG	15 min	Telekurs
Germany	XETRA Stock Exchange	.DE	15 min	Telekurs
Hong Kong	Hong Kong Stock Exchange	.HK	15 min	Comstock
India	Bombay Stock Exchange	.BO	15 min	Comstock
India	National Stock Exchange of India	.NS	15 min**	Exchange of India
Indonesia	Jakarta Stock Exchange	.JK	10 min	Comstock
Israel	Tel Aviv Stock Exchange	.TA	20 min	Telekurs
Italy	Milan Stock Exchange	.MI	20 min	Telekurs
Korea	Korea Stock Exchange	.KS	20 min	Comstock
Korea	KOSDAQ	.KQ	20 min	Comstock
Mexico	Mexico Stock Exchange	.MX	20 min	Telekurs
Netherlands	Amsterdam Stock Exchange	.AS	15 min	Telekurs
New Zealand	New Zealand Stock Exchange	.NZ	20 min	Comstock
Norway	Oslo Stock Exchange	.OL	15 min	Telekurs
Singapore	Singapore Stock Exchange	.SI	20 min	Comstock
Spain	Barcelona Stock Exchange	.BC	15 min	Telekurs
Spain	Bilbao Stock Exchange	.BI	15 min	Telekurs

续表

国家及地区	交易所名称	股票后缀	行情延迟时间	数据来源
Spain	Madrid Fixed Income Market	.MF	15 min	Telekurs
Spain	Madrid SE C.A.T.S.	.MC	15 min	Telekurs
Spain	Madrid Stock Exchange	.MA	15 min	Telekurs
Sweden	Stockholm Stock Exchange	.ST	15 min	Telekurs
Switzerland	Swiss Exchange	.SW	30 min	Telekurs
Taiwan	Taiwan OTC Exchange	.TWO	20 min	Comstock
Taiwan	Taiwan Stock Exchange	.TW	20 min	Comstock
United Kingdom	London Stock Exchange	.L	20 min	Telekurs

附录 7

部分美国股票代码表

代　码	公司名称	代　码	公司名称
AAI	AirTran Holdings Inc.	KMR	Kinder Morgan Management, LLC
AAPL	Apple Inc	KMX	Carmax Inc
ABC	AmerisourceBergen Corp	KNXA	Kenexa Corp
ABM	ABM Industries Inc.	KO	Coca Cola Co
ABX	Barrick Gold Corp	KR	Kroger Co
ABY	Abitibi—Consolidated Inc	LAYN	Layne Christensen Co
ACLS	Axcelis Technologies Inc	LCAPA	Liberty Media Hldg Corp (Cap)
ACS	Affiliated Computer Services	LCC	U S Airways Group Inc
ACV	Alberto Culver Co New	LDG	Longs Drug Stores Corp
ADCT	A D C Telecommunications	LEE	Lee Enterprises Inc
ADI	Analog Devices Inc.	LEG	Leggett & Platt
ADM	Archer Daniels Midland Co	LGF	Lions Gate Entmnt Corp
ADTN	Adtran Inc	LH	Laboratory Corporation of America
ADVS	Advent Software, Inc.	LINTA	Liberty Media Hldg Corp
AEOS	American Eagle Outfitters New	LIZ	Liz Claiborne
AES	Aes Corp	LM	Legg Mason Inc
AFFX	Affymetrix Inc	LMT	Lockheed Martin Corp
AFL	Aflac Inc	LNC	Lincoln National Corp.

续表

代　码	公司名称	代　码	公司名称
AG	Agco Corp	LNOP	Lanoptics Ltd
AGIL	Agile Software Corp Del	LOW	Lowes Companies
AGRA	Agere Sys Inc	LRCX	Lam Research Corp
AHG	Apria Healthcare Group Inc	LSCC	Lattice Semiconductor Corp
AIV	Apartment Investment and Management	LSI	Lsi Logic Corp
ALC	Assisted Living Concpt Nev New Cl A	LTD	Limited Brands, Inc.
ALTH	Allos Therapeutics Inc	LTM	Life Time Fitness Inc
ALV	Autoliv Inc.	LVLT	Level 3 Communications Inc
AMAT	Applied Matls Inc	LVS	Las Vegas Sands Corp
AMCC	Applied Micro Circuits Corp	LYTS	Lsi Inds Inc
AMD	Advanced Micro Devices Inc	LYV	Live Nation Inc
AMG	Affiliated Managers Group Inc	MAN	Manpower Inc
AMP	Ameriprise Financial Inc	MAR	Marriott Intl Inc New
AMT	American Tower Corp.	MAT	Mattel Inc.
AMWD	American Woodmark Corp	MBRX	Metabasis Therapeutics Inc
AMZN	Amazon Com Inc	MBT	Mobile Telesystems Ojsc
ANDW	Andrew Corp	MCD	Mcdonalds Corp
ANF	Abercrombie & Fitch Co	MCO	Moodys Corp
AOI	Alliance One Intl Inc	MCY	Mercury General Corp.
ARB	Arbitron Inc	MEDI	MedImmune Inc.
ARC	Affordable Residential Cmtys	MEH	Midwest Air Group Inc
ARG	Airgas Inc	MEL	Mellon Finl Corp
ARIA	Ariad Pharmaceuticals Inc	MERCS	Mercer Intl Inc
ARO	Aeropostale	MET	Metlife Inc
ARXT	Adams Respiratory Therapeuti	MFLX	Multi Fineline Electronix Inc
ASD	American Standard Companies	MGM	Mgm Mirage
ASGN	On Assignment Inc	MHP	Mcgraw Hill Cos Inc
ASN	Archstone Smith Tr	MIR	Mirant Corp New
ASTSF	Ase Test Ltd	MMM	3M Co
AT	Alltel Corporation	MNS	Msc Software Corp
ATR	Aptargroup Inc	MNST	Monster Worldwide Inc

续表

代　码	公司名称	代　码	公司名称
ATU	Actuant Corp	MO	Altria Group Inc
AU	Anglogold Ashanti Ltd	MON	Monsanto Co New
AUXL	Auxilium Pharmaceuticals Inc	MOS	Mosaic Co
AVID	Avid Technology, Inc.	MOT	Motorola Inc
AVNX	Avanex Corp	MRVL	Marvell Technology Group Ltd
AVP	Avon Products, Inc.	MS	Morgan Stanley
AVY	Avery Dennison Corp	MSA	Mine Safety Appliances Co.
AVY	Avery Dennison Corporation	MSFT	Microsoft Corp
AVZ	AMVESCAP PLC	MTB	M & T Bank Corp
AW	Allied Waste Inds Inc	MU	Micron Technology Inc
AW	Allied Waste Industries Inc	MVL	Marvel Entertainment Inc
AXP	American Express Co	MW	Mens Wearhouse Inc
AXR	Amrep Corp New	MXIM	Maxim Integrated Prods Inc
AYI	Acuity Brands Inc	MYGN	Myriad Genetics Inc
BA	Boeing Co	NAK	Northern Dynasty Minerals Ltd
BARE	Bare Escentuals Inc	NCR	Ncr Corp New
BAX	Baxter International, Inc.	NDSN	Nordson Corp
BBA	Bombay Co Inc	NEM	Newmont Mining Corp
BBBY	Bed Bath & Beyond Inc	NEU	Newmarket Corp
BBBY	Bed, Bath & Beyond, Inc.	NFLX	Netflix Com Inc
BBD	Banco Bradesco S A	NKE	Nike Inc
BBI	Blockbuster Inc	NKTR	Nektar Therapeutics
BBND	Bigband Networks Inc	NLY	Annaly Capital Management, Inc
BBW	Build A Bear Workshop	NMX	Nymex Holdings Inc
BBY	Best Buy Inc	NOK	Nokia Corp
BCR	C. R. Bard, Inc.	NOK	Nokia Oyj
BE	Bearingpoint Inc	NOVL	Novell Inc
BEAS	BEA Systems Inc.	NP	Neenah Paper Inc
BEAV	BE Aerospace Inc	NPSP	Nps Pharmaceuticals Inc
BEBE	Bebe Stores Inc	NRG	Nrg Energy Inc
BEC	Beckman Coulter, Inc.	NSC	Norfolk Southern

续表

代　码	公司名称	代　码	公司名称
BECN	Beacon Roofing Supply Inc	NSC	Norfolk Southern Corp
BEN	Franklin Res Inc	NTAP	Network Appliance Inc
BEZ	Baldor Elec Co	NTRI	Nutri Sys Inc New
BFLY	Bluefly Inc	NUE	Nucor Corp
BG	Bunge Ltd.	NVL	Novelis Inc
BID	Sothebys	NWL	Newell Rubbermaid Inc
BIG	Big Lots Inc	NWS	News Corp
BIVNE	Bioenvision Inc	NWY	New York & Co Inc
BLC	Belo Corp	OEH	Orient－Express Hotels Ltd
BLS	BellSouth Corp.	OIH	Oil Svc Holdrs Tr
BLT	Blount International Inc.	OLN	Olin Corp
BMC	Bmc Software Inc	OMG	Om Group Inc
BMS	Bemis Company, Inc.	OPSW	Opsware Inc
BMTI	Biomimetic Therapeutics Inc	ORCL	Oracle Corp
BNI	Burlington Nothern Santa Fe	PAAS	Pan American Silver Corp
BOBJ	Business Objects S A	PALM	Palm Inc New
BONT	Bon－Ton Stores Inc	PBI	Pitney Bowes Inc
BORL	Borland Software Corp	PBR	Petroleo Brasileiro Sa Petrobr
BOW	Bowater Inc	PCLN	Priceline Com Inc
BRCD	Brocade Communications	PCU	Southern Copper Corp
BRCM	Broadcom Corp	PDE	Pride Intl Inc Del
BRKS	Brooks Automation Inc	PDLI	Pdl Biopharma Inc
BRO	Brown & Brown Inc.	PERY	Ellis Perry Intl Inc
BTRX	Barrier Therapeutics Inc	PETM	PetSmart, Inc.
BUCY	Bucyrus Intl Inc New	PETS	Petmed Express Inc
BUD	Anheuser－Busch Cos. , Inc.	PFE	Pfizer Inc
BWLD	Buffalo Wild Wings Inc	PG	Proctor & Gamble
C	Citigroup Inc	PGN	Progress Energy Inc.
CA	Ca Inc	PGR	Progressive Corp Ohio
CAB	Cabelas Inc	PII	Polaris Industries Inc.
CAE	Cascade Corp	PIR	Pier 1 Imports

续表

代　码	公司名称	代　码	公司名称
CALP	Caliper Life Sciences Inc	PLCE	Childrens Pl Retail Stores Inc
CAMP	Calamp Corp	PLXT	Plx Technology Inc
CASY	Caseys Gen Stores Inc	PMCS	Pmc－Sierra Inc
CAT	Caterpillar Inc Del	PNRA	Panera Bread Co
CBK	Christopher & Banks Corp	POL	Polyone Corp
CBRL	CBRL Group Inc	POS	Catalina Marketing Corp
CBT	Cabot Corp.	POT	Potash Corp Sask Inc
CCC	Calgon Carbon Corp	PPG	P P G Industries
CCL	Carnival Corporation	PPL	Ppl Corp
CDCO	Comdisco Holdings	PRU	Prudential Finl Inc
CEDC	Central European Dist Corp	PSS	Payless Shoesource Inc
CENX	Century Alum Co	PSUN	Pacific Sunwear Calif Inc
CEPH	Cephalon Inc	PTR	Petrochina Co
CERN	Cerner Corp	PTRY	Pantry Inc
CHAP	Chaparral Stl Co Del	PTV	Pactiv Corp.
CHIC	Charlotte Russe Hldg Inc	PVH	Phillips Van Heusen Corp
CHKP	Check Point Software Tech Ltd	PWER	Power－One Inc
CHP	C&D Technologies Inc	PX	Praxair Inc
CHTT	Chattem Inc	PXLW	Pixelworks Inc
CINF	Cincinnati Financial Corp.	QCOM	Qualcomm Inc
CL	Colgate Palmolive Co	RAD	Rite Aid Corp
CLE	Claire's Stores, Inc.	RAH	Ralcorp Hldgs Inc New
CLF	Cleveland Cliffs Inc	RAI	Reynolds American
CLG	Cumberland Res Ltd	RAIL	Freightcar Amer Inc
CMA	Comerica Incorporated	RBN	Robbins & Myers Inc
CMCSA	Comcast Corp New	RCII	Rent A Ctr Inc New
CMCSK	Comcast Corp	RCII	Rent－A－Center Inc.
CME	Chicago Mercantile Hldgs Inc	RCL	Royal Caribbean Cruises Ltd.
CMG	Chipotle Mexican Grill Inc (Class A)	RDWR	Radware Ltd
CMRG	Casual Male Retail Grp Inc	RDYN	Replidyne Inc
CNO－B	Conseco Inc	RE－A	Everest Re Group Ltd.

续表

代 码	公司名称	代 码	公司名称
COGN	Cognos Inc.	RF	Regions Financial Corp.
COH	Coach Inc	RGS	Regis Corporation
COO	The Cooper Companies Inc.	RHI	Robert Half Intl Inc
COP	ConocoPhillips	RI	Ruby Tuesday, Inc.
COST	Costco Wholesale	RIG	Transocean Inc
CPHD	Cepheid	RIO	Companhia Vale Do Rio Doce
CPWR	Compuware Corp	RKT	Rock－Tenn Co
CRDN	Ceradyne Inc	RL	Polo Ralph Lauren Corp
CRI	Carters Inc	RMIX	U S Concrete Inc
CRL	Charles River Laboratories Int	RNT	Aaron Rents Inc
CROX	Crocs Inc	ROH	Rohm & Haas Co.
CSC	Computer Sciences Corp	ROL	Rollins Inc
CSCO	Cisco Sys Inc	ROST	Ross Stores, Inc.
CSG	Cadbury Schweppes Plc	RPM	RPM International, Inc.
CSGS	CSG Systems International, Inc	RRD	R. R. Donnelley & Sons Company
CSX	Csx Corp	RSG	Republic Svcs Inc
CTAS	Cintas Corp	RSG	Republic Services, Inc.
CTR	Cato Corp, The	RSTO	Restoration Hardware Inc Del
CTRN	Citi Trends Inc	RSYS	Radisys Corp
CTXS	Citrix Sys Inc	RSYS	Radisys Corp
CVS	Cvs Caremark Corporation	RTI	Rti Intl Metals Inc
CVX	Chevron Corp.	RVI	Retail Ventures Inc
CWTR	Coldwater Creek Inc	SAI	Saic Inc
CX	Cemex Sab De Cv	SBGI	Sinclair Broadcast Group Inc
CY	Cypress Semiconductor Corp	SBH	Sally Beauty Hldgs Inc
DADE	Dade Behring Hldgs Inc	SBUX	Starbucks Corp
DAKT	Daktronics Inc	SCHL	Scholastic Corp
DBD	Diebold, Incorporated	SCHN	Schnitzer Steel Industries Inc
DBRN	Dress Barn Inc	SCHW	Schwab Charles Corp New
DD	Du Pont E I De Nemours & Co	SEE	Sealed Air Corp.
DGX	Quest Diagnostics Inc	SEH	Spartech Corp.

目　　录